CONTENT-BASED MUSIC CLASSIFICATION, SUMMARIZATION AND RETRIEVAL

SHAO XI

NATIONAL UNIVERSITY OF SINGAPORE

2006

CONTENT-BASED MUSIC CLASSIFICATION, SUMMARIZATION AND RETRIEVAL

SHAO XI
(B. Eng, M. Eng)
Nanjing University of Posts and Telecommunications

A DISSERTATION

SUBMITTED TO SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

SCHOOL OF COMPUTING

NATIONAL UNIVERSITY OF SINGAPORE

Acknowledgements

I would like to express my deep and sincere gratitude to my supervisor, Professor Mohan S. Kankanhalli, his wide knowledge and his logical way of thinking have been of great value for me. His understanding, encouraging and personal guidance have provided a good basis for the present thesis and will benefit me all through my life.

I would owe my warm and sincere thanks to my supervisor in Institute for Inforcomm Research, Dr. Xu Changsheng, who gave me important guidance during my first steps into this research area. Thank for his detailed and constructive comments, and for his important support throughout this work.

I owe my loving thanks to my family. They have lost a lot due to my research abroad. Without their encouragement, endurance and understanding it would have been impossible for me to finish this work. My special gratitude is due to my parents for their support. I would like to share this moment of happiness with them.

The episode of acknowledgement would not be complete without the mention of my colleagues in Institute for Inforcomm Research, Singapore. Namunu, Jinjun, Yantao, Qiubo, Lingyu, and thanks for their friendly help and social support during the period of my graduate study.

Finally, I would like to thank all the people who directly or indirectly gave me support to help me complete my thesis in time, and thank to Institute for Inforcomm Research for providing me a nice research environment.

Contents

Summary

With the explosive amount of music data available on the internet in recent years, there has been a compelling need for the end user to search and retrieve effectively in increasingly large digital music collection. In order to manage the real-world digital music database, some applications are needed to help people manipulate the large digital music database.

In this work, three issues in real world digital music database management were tackled. These issues include music summarization, music genre classification and music retrieval by human humming, as these three applications satisfy the basic requirement of an operational real world music database management system. Among these three applications, music genre classification and music summarization perform music analysis and find the structure information both for the individual songs in database and the whole music database, which can speed up the searching process, while music retrieval is an interactive application. In this thesis, these issues were addressed using machine learning approaches, complementary to digital signal processing method. To be specific, the digital signal processing helps extract compact, task dependent information-bearing representation from raw acoustic signals, i.e., music summarization and classification employ timber features and rhythm features to characterize the music content, while music retrieval by humming requires the melody features to characterize the music content. Machine learning includes segmentation, classification, clustering and similarity measuring, etc., and it pertains to computer understanding of the music contents. We proposed an adaptive clustering approach for

structuring the music content in music summarization, extended the current music genre classification by a supervised hierarchical classification approach and an unsupervised classification approach, and in query by humming, in order to separate the vocal content from the polyphonic music, we proposed a statistical learning based method to solve the permutation inconsistency problem for Frequency-Domain Independent Component Analysis. In most cases, the proposed algorithms for these three applications have been evaluated by conducting user studies, and the experimental results indicated the proposed algorithms were effective in helping realize users' expectations in manipulating the music database.

In general, since the semantic gap exists between low level representation of music signals and different level applications in music database management, machine learning is indispensable to bridge such gap. Furthermore, machine learning approach can also be incorporated into signal processing to solve difficult problems.

List of Tables

List of Figures

Introduction 1

The rapid development of affordable technologies for multimedia content capture, data storage, high bandwidth/speed transmission and for multimedia compression, have resulted in a rapid increase of the size of digital multimedia data collections and greatly increased the availability of multimedia content for the general users. However, how to manage and interact with the ever increasing multimedia database has become an increasingly important issue for these users. One of the most practical ways to solve this problem relies on multimedia database management which aims to search and retrieve user required parts of multimedia information stored in the database.

Music is one of the most important media types intimately related to our lives. The penetration of music technology has progressed to the point that today comparatively few households are without digital music in the form of compact discs, mini-discs or MP3 players. The ubiquity of digital music is further evidenced by the multimedia capabilities of the modern personal computer and by the high speed transmission of Internet. 10,000 new albums are released and 10,000 works registered

for copyright in 1999 [1], and for US alone, 420 million recorded music (e.g. CDs, cassettes, music videos and so forth), were downloaded and recorded company revenues an estimated US$ 1.1 Billion in 2005 [2]. Therefore, there is a compelling need for the end user to search and retrieve effectively in increasing large digital music collections. Most existing music searching tools build upon the success of text search engines (i.e. www.google.com , www.1sou.com, etc.), which operate only on the annotated text metadata. However, they become non-functional when meaningful text descriptions are not available. Furthermore, they do not provide any means to search on the music content.

A truly content based music information retrieval system should have the ability to manage music information based on their content [3], other than the text metadata. Traditional techniques used in text searching do not easily carry over to the music domain, and new technology needs to be developed. Before developing the new technology for music information systems, at first we should take a look at background of current technology for music library management systems.

1.1 Background

For comparison purposes, we would like to link the concept of music library management with the concept in the traditional management of book library. In Figure 1-1, the left figure shows the paradigm in the traditional management of book library and the right figure shows the paradigm of music library management. In the management of book library, the on-shelf books are first classified into different

categories to facilitate the retrieval process. For each particular book, the table of contents serves as an index to different sections of the book and the abstract serves as the overview of the whole book. The table of contents and abstract of the book also aim to help users efficiently access just the required parts of information.

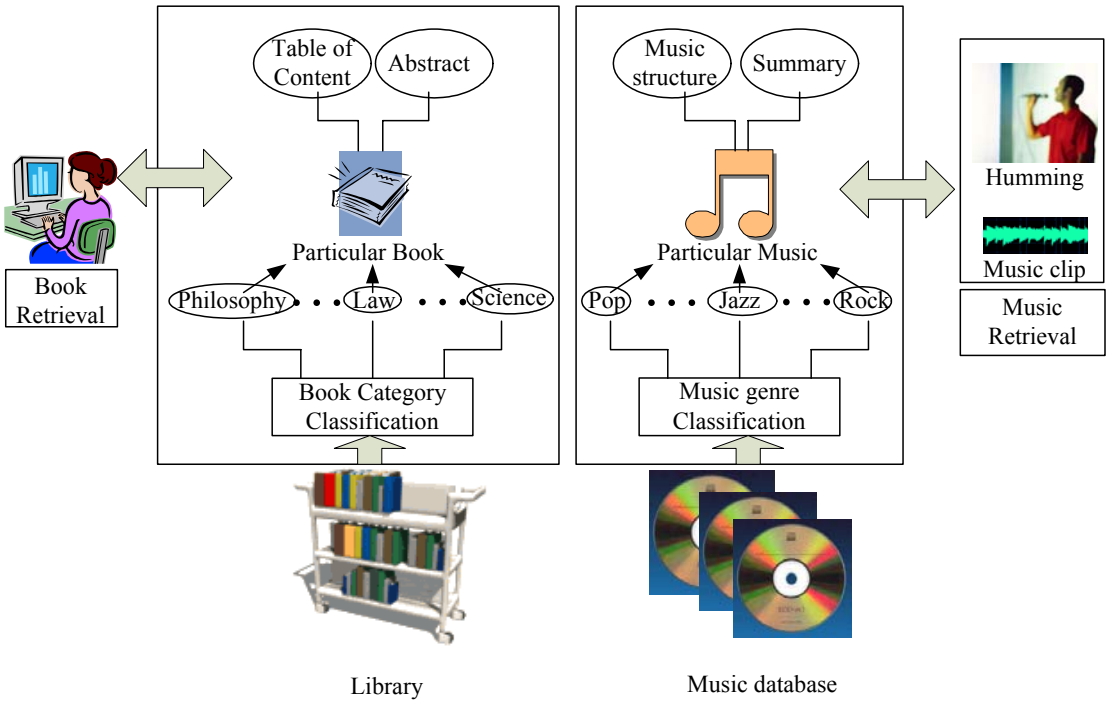


Figure 1-1: The concept paradigm of music database management and traditional management of book library

Similarly, an analogous concept can be used for music library management, as the figure shows in the right side. The music genre classification module takes the role of book classification in the traditional book library management and categorizes each music piece according to its inherent genre identification. As for each music piece, to efficiently access just the required parts of music content, we can index the music content by music structure analysis, and we can also give the users the main theme of the music work by showing them a shorter music summary condensed from the original music. After the music database has been structured, searching and retrieving in the music library management will be easy and efficient, as long as the

functionality of all the modules in the book library management can be realized in the music library management. However, in book library management, the database side and query side are all text-based and almost all text information retrieval methods which rely on identifying approximate units of semantics, that is, words, can be applicable. In music library management, locating such units in the database side is extremely difficult, perhaps impossible, since the database side is raw music signal. A natural and direct solution for music library management is to index the music content using textual descriptions. But this has the problem of subjectivity, as it is hard to have a "generic" way to first describe and then retrieve music content that is universally acceptable. This is inevitable as users interpret semantics associated with the music content in so many different ways, depending on the users, the purpose of use, and the task that needs to be performed. The problem gets even murkier, as the purpose for retrieval is often completely different from the purpose for which the content was created, annotated and stored in the database. For example, the query side usually may not have the same representation as that in the database side (i.e. humming-based query, text-based query). The heterogeneity of two entities in database side and query side has been proven to be the source of the most intractable problems in music information retrieval [4].

Alternatively, another solution to music database management is to index the music database by the content itself, which has received a lot of attention in recent years [5][6][7][8]. Content refers to any information about music signal and its structure. Some examples of content information are: knowing a specific section of a

song corresponding to the verse or chorus, identifying the genre information of a specific music piece, etc. The content based music information retrieval start with techniques that could automatically index the music content based on some inherent features, such kind of features could be extracted from music content itself. For example, features such as rhythms, tonality, timbres, etc., can be easily extracted from raw music signals using current techniques of digital signal processing. As a result, the content based music library management can partially avoid the problems caused by textual labeling based music library management; however, such features have proven to be inconsistent with human perception of the music work [9]. Especially in the retrieval process, as the query is generated from the view point of human perception, which is more abstract and subjective than what the low level features can express.

Therefore, in content based music library management, the low level features cannot provide sufficient information for retrieval. Between low-level features and the applications in music database management, there is a semantic gap which corresponds to human understanding of the music content. In order to retrieve music information more effectively, we need to go deeper into the music content and exploit the semantics from the viewpoints of human perception of music, where the focus has been in understanding inherent digital signal characteristics that could offer insights into semantics situated within the music content. The major work in this thesis is to show that machine learning plays a fundamental role for the applications of different levels in the music database management, complementary to digital signal processing.

1.2 Main Problem Statement

The main problem that our work tries to address is *the use of digital signal processing methods, combined with machine learning approach, for several applications in real world digital music database management.* To be specific, these applications include music summarization, music genre classification and music retrieval by humming. Among them, two are middle level applications and one is the high level interactive application. The interactive application refers to music retrieval by humming, and two middle level applications include music genre classification and music summarization.

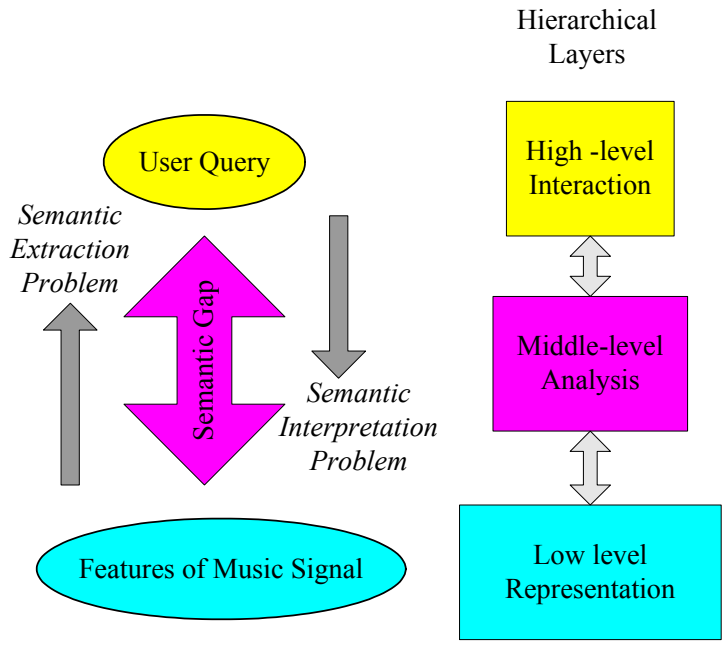


Figure 1-2: The hierarchical structure for music database management system

The relationship between three applications can be illustrated in Figure 1-2. Music genre classification and music summarization correspond to the music analysis stage in hierarchical structure of music database management. These two applications

perform music analysis and find the structure information both for the individual songs in database and the whole music database, which can speed up the searching process, while music retrieval is a high level interactive application, corresponding to interaction stage in Figure 1-2.

As the Figure1-2 shows, from the bottom up manner, low level feature representation of the music signal lies at the bottom of the hierarchical structure in music database management system. One single feature objectively reflects one or some perceptually relevant aspects of the music content. For example, the rhythm features carry the tempo information of the music content, while the timbre features carry the texture information of the music content. Once the features have been correctly extracted from the raw music signals, they can be considered as physical parameters measuring one or some aspects of the raw music signals. A drawback, however, is that these low-level features are often too restricted to describe the music content on a conceptual or semantic level. As the stage of music database management hierarchical system goes up, the music content needs to be interpreted more subjectively. In analysis stage, music database management should have the self-organized ability according to the semantic understanding of the low level features. For example, to organize the music database efficiently, we need classify each song into different genre according to the genre information it carries. However, the perceptual criteria of music genre is not only related to the low level features such as melody, tempo, texture, instrumentation and rhythmic structure, but also an intuitive concept determined by people's understanding of the particular songs. In the

interaction stage; the user generates the query from the view point of human perception. Take the query by humming as an example: the users are most likely to hum a few memorable bars which are usually the most salient part of the music. If we can locate such salient part in each music piece, not only the searching space will be reduced, but also the retrieval accuracy will be improved. However, the low level feature cannot directly provide such kind of conceptual information. Therefore, from the bottom up manner in hierarchical structure, there is a problem so-called *semantic extraction problem*. In top-down manner, the query generated from human being is subjective and arbitrary, i.e. a humming contains variation and inaccuracy, and how to interpret such kind of query to objective low level representation is not trivial. This is the so-called *semantic interpretation problem*. Therefore, between low level features and high level interactive applications, there is a semantic gap which corresponds to human understanding of the music contents. It is our opinion that ignoring the existence of the semantic gap was the cause of many disappointments in the performance of early music database management.

To summarize, digital signal processing play the important role in real world music database management, since various low level features should be accurately extracted from the raw music signals using digital signal processing methods. Complementary to digital signal processing, machine learning plays a fundamental role in real world music database management. Without music semantic understanding, middle level and top level applications in music database management will be very difficult to handle, or even impossible to handle. The machine learning

approaches, by providing semantic understanding for music database, can bridge the gap between low level features and different level applications in music database management.

One of the conceptual architectures for content based music database management is shown in Figure 1-3. In this illustration, the rectangle represents the procedure/method that needs to be designed and developed, and the rectangle with rounded corners represents the out entity or result from the system.

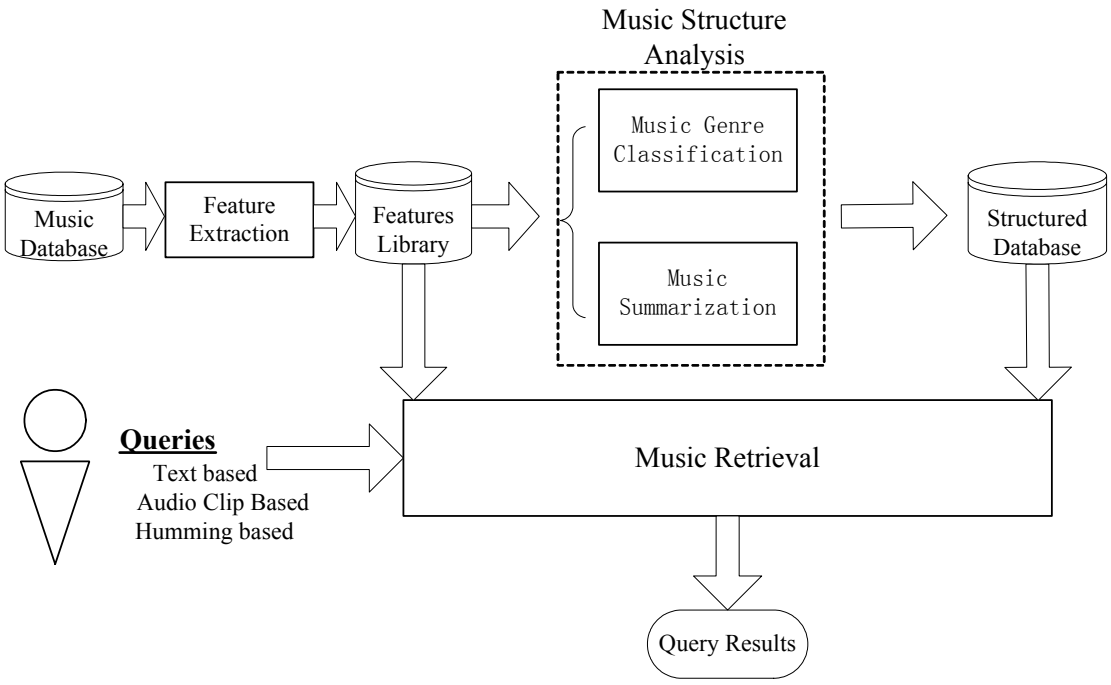


Figure 1-3: The architecture of content based music database management

Firstly, the feature extraction procedure is applied on the music database which contains various types of real world audio files, such as .wav format. After feature extraction, we gather these features to build the feature library. This procedure corresponds to the audio representation stage in Figure 1-2.

Once the features have been extracted, music structure information both for music

database and for each music piece in the database side should be obtained by various machine learning approaches. It actually partitions the music in database in two orientations: “vertical” orientation and “horizontal” orientation. In “vertical” orientation, music genre classification partitions the music pieces in database according to their inherent genre identification. In the context of large musical databases, genre is therefore a crucial metadata for the description of music content. While in “horizontal” orientation, music summarization structures the individual music piece in database according to its intrinsic repeating patterns and the role which these segments play in the whole music. The aim of music summarization is to choose the most representative segment (or segments) to represent the whole music, using the music structure information. It can provide the entry for the most repeated parts of the music. These repeating patterns and structure of the individual music piece are very helpful in music database management, since such kind of representative segments contain most memorable information for human beings and in the retrieval process, giving the high priority to these segments will significantly reduce the searching space. As a result, interaction with large music database can be made simpler and more efficient.

Finally, a polyphonic music retrieval mechanism can be built based on the archiving scheme describe above. Search queries might be constructed using a variety of input method. These may include: manual editing within a graphical or textual dialog; the music clips; or even whistling, humming into a microphone. We focus our research on query by humming since humming is most natural way to formulate

music queries for people who are not trained or educated with music theory .The music retrieval procedure corresponds to the audio interaction stage in Figure 1-2.

The conceptual architecture of content based music database management described above has a hierarchical and modular structure in which the physical and perceptual natures of different types of music are well organized. It is flexible in the sense that each layer/module may be developed individually and has its own application domain. It should be noted that conceptual architecture described is just a general architecture for content based music database management. Under this architecture, a lot of work can be combined into this framework both on database and query. We try to address three main applications in this architecture, which include music genre classification, music summarization and music retrieval via query-by-humming on real world music database, using digital signal processing methods, combined with machine learning approaches. In addition, we choose the polyphonic music representation in the database side since it constitutes the bulk of the real world audio files. Polyphonic music is much more prevalent in the real world than the monophonic music representation.

1.3 Concept Linkage between Three Applications

It also should be noted that the music genre classification, music summarization and music retrieval are not isolated, and the success of one aspect will contribute to the others.

Firstly, the results of music summarization and music genre classification will be

helpful to each other. On one hand, music structure information can be utilized in music genre classification. For example, some music genres have a fairly rigid format, others are more flexible. Therefore, using the music structure information, we can roughly classify the music genre at a coarse-level. On the other hand, the aim of music summarization is to choose the most representative segment (or segments) to represent the whole music, using the music structure information. Since different music genres have a different music structure and the most representative part for each genre relies on its own intrinsic distinctive portion, it is essential to classify a music piece into a certain genre category before employing a genre-specific summarization schemes. For example, the most distinguished portion of Pop music is the chorus, which repeats itself several times in the whole music structure, while for Hip-Hop music, there is no such repetition, and the music summarization approach for Hip-Hop music would be different from the one for Pop music.

Secondly, music genre classification would be helpful for music retrieval. With the aid of music genre classification, the music retrieval process would be more efficient and effective. For example, for the user query, if we can recognize the music genre information of the humming query, which is provided by music genre classification model, and then the search space of the target melody can be limited to the music titles of the certain genre in the database. As a result, the search space for the retrieval will be significantly reduced. In addition, musical content features that are good for genre classification can be used in other types of analyses such as similarity retrieval, because they do carry a certain amount of genre-identifying

information and therefore are a useful tool in content–based music analysis.

Thirdly, music summarization would constitute a valuable addition to music retrieval. One could, for instance, hum a few memorable bars to formulate music queries. This query melody can begin at any instant of a song. To find the target melody, we need to search the each song thoroughly in the huge music database, which is time consuming and not practical, or even impossible, for real world applications. However, with the aid of music summarization result, the retrieval process will be simpler and more efficient. This is because a music layman is most likely to hum a few memorable bars which fall in the most repeated part of a song. In this way, the database side of the retrieval system can be focused on the music summary, instead of the original song. Thus, it can serve as a filtering mechanism. On the other hand, from the human computer interaction viewpoint, music summarization is important for music retrieval especially for the presentation of the returned ranked list since it allows users to quickly hear the results of their query and make their selection.

Finally, to make MIR in real sound recording more practical, information from different sections such as instrumental setup, rhythm, melody contours, key changes and multi source vocal information in the song needs to be extracted. Organizing such information is challenging but possible with structural analysis provided by music summarization and classification.

1.4 Main Contributions

In this section, the main contributions in this thesis are briefly reviewed. More details and explanation of the terms will be provided in the succeeding chapters.

- Extension of current prescriptive approach and emergent approach for music genre classification

A hierarchical classifier based on SVM was proposed to discriminate musical genres, which extend the current prescriptive approach for music genre classification not only reducing the complexity of each single task, but also improving the global classification accuracy.

For emergent approach, Hidden Markov Models (HMMs) were employed to model the relationship between features over time from the raw songs. As a result, the similarity of each song in music collections can be measured using the distance provided by the HMMs. Based on the song similarities, an un-supervised clustering method can be used to emerge the music genres.

- Adaptive clustering algorithm in music summarization

We propose adjusting of the overlapping rate of the music signal segmentation window, which aims to optimally group the music frames to get the good summarization results.

- Audio-visual alignment algorithm for music video summarization

Based on summary for music track, we propose the structuring of the visual content, followed with visual and audio alignment to generate an audio/video summary for music videos which maximizes the coverage of important audio

segments along with important video segments.

- Statistical learning approach to solve the *permutation inconsistency* problem in Frequency Domain Independent Component Analysis(FD-ICA)

Considering the vocal singing voice and background music as two heterogeneous signals, we present a predominant vocal content separation method for two -channel polyphonic music by employing a statistical learning based method to solve the *permutation inconsistency* problem in FD-ICA.

1.5 Thesis Overview

This thesis is organized as follows:

Chapter 1 (which you are currently reading) provides an overview of the whole thesis, including the introduction to the background of the music database management, main problem this thesis tries to address, and main contributions our work has achieved.

In Chapter 2, we present two approaches for automatically classifying music genres, one is based on supervised learning and the other is based on unsupervised learning.

In Chapter 3, we have proposed a summarization approach which extracted the most salient part of music based on adaptive clustering, with the help of music structure analysis. In addition, we also extended our proposed music summarization to the music video summarization scheme.

In Chapter 4, we present a practical query by humming music retrieval system for real world music database. As an extension of query by humming music retrieval system for monophonic music database, the most difficulty in query by humming music retrieval system for real world music database is how to separate one monophonic representation from the polyphonic music. In this chapter, we present a predominant vocal content separation method for two-channel polyphonic music by employing a statistical learning based method, combined with the signal processing approach.

The experimental results of our proposed music database structuring and retrieval algorithms are described and discussed in Chapter 5. The thesis ends with Chapter 6 which summarizes the whole thesis and gives directions for future research.

2 Music Genre Classification

Music genre classification is a middle level application for music database management. It partitions the music pieces in database according to their inherent genre identification. In the context of large musical databases, genre is therefore a crucial metadata for the description of music content. The ever increasing wealth of digitized music on the Internet, music content in digital libraries and peer to peer systems call for an automated organization of music materials, as it is not only useful for music indexing and content-based music retrieval, but also can be used for other middle level music analysis applications such as music summarization. Although to make computers understand and classify music genre is a challenging task, with the help of machine learning approaches, there are still perceptual criteria related to the melody, tempo, texture, instrumentation and rhythmic structure that can be used to characterize and discriminate different music genres.

2.1 Related Work

A music genre is characterized by common features related to instruments, texture,

dynamics, rhythmic characteristics, melodic gestures and harmonic content. The first challenge of genre classification is to determine the relevant features and find a way to extract them.

2.1.1 Feature Extraction

Since the low level audio samples contain low 'density' of the information, they cannot be directly used by an automatic analysis system. Therefore, the first step of analysis systems is to extract some features from the audio data to manipulate more compact information from raw audio signal. In the case of the music genre classification, features may be related to the main dimensions of music genres including timbre, harmony, and rhythm.

A. Timbre Features:

Timbre is defined in literature as the perceptual feature that makes two sounds different with the same pitch and loudness [10]. Features characterizing timbre analyze the spectral distribution of the signal though some of them are computed in the time domain. These features are global in the sense that integrates the information of all sources and instruments at the same time.

An exhaustive list of features used to characterize timbre of the music can be found in [11]. Here, we summarize the main timbre features used in genre characterization:

- **Temporal features**: features i.e., zero-crossing rate [12], linear prediction coefficients [12], etc.

- **Spectrum shape features:** features describing the shape of the power spectrum of a signal frame. i.e., Spectral centroid[13], spectral rolloff[14], spectral flux[14], octave-based spectral contrast feature[15], MFCCs[12], etc.
- **Energy features:** features referring to the energy content of the signal. i.e., Root Mean Square energy of the signal frames, energy of the harmonic component of the power spectrum, etc.

Transformations of features such as first and second-order derivatives are also commonly used to create new features for the purpose of modeling the dynamic property of the music signals.

B. Melody features

Melody is a succession of pitch events perceived as a single entity. Pitch is a perceptual term which can be approximated by fundamental frequency. The pitch content features describe the melody and harmony information about music signals and pitch content feature set is extracted based on various multi-pitch detection techniques. A good overview of melody description and extraction in the context of audio content processing can be found in [16]. At the current stage it is only possible to determine the real pitch of every note of monophonic signals, but not from polyphonic complex music. Therefore, the pitch related features usually only estimate the distribution of peaks in the frequency spectrum by determining them directly by autocorrelation. For example, the multi-pitch detection algorithm described in [17] can be used to estimate the pitch. In this algorithm, the signal is decomposed into two frequency bands and an amplitude envelope is extracted for each frequency band. The

envelopes are summed and an enhanced autocorrelation function is computed so that the effect of integer multiples of the peak frequencies on multiple pitch detection is reduced. The prominent peaks of this summary enhanced autocorrelation function correspond to the main pitches for that short segment of sound and are accumulated into pitch histograms. Then, the pitch content features can be extracted from the pitch histograms.

C. Rhythm features

Rhythmic features characterize the movement of music signals over time and contain information such as the regularity of the rhythm, beat, tempo, and time signature. A review of automatic rhythm description systems may be found in [18].These automatic systems may be oriented towards different applications: tempo induction, beat tracking, meter induction, or quantization of performed rhythm. However, the current rhythm description systems still have a number of weaknesses, so that they do not give reliable information for machine learning algorithm. In light of this, a descriptor measuring the importance of periodicities in the range of perceivable tempo (typically 30-200 Mälzel's Metronome) should be obtained in a statistical manner. Such descriptor for representing rhythm structure is usually extracted from the beat histogram. Tzanetakis [19] used a beat histogram built from the autocorrelation function of the signal to extract rhythmic content features. The time-domain amplitude envelopes of each band are extracted by decomposing the music signal into a number of octave frequency bands. Then, the envelopes of each band are summed together followed by autocorrelation of resulting sum envelopes.

The dominant peaks of the autocorrelation function, corresponding to the various periodicities of signal's envelopes, are accumulated over the whole music source into a beat histogram where each bin corresponds to the peak lag.

D. Wavelet features

The Wavelet Transform (WT) is a technique for analyzing signals. It was developed as an alternative to Short Time Fourier Transform (STFT) to overcome the problem related to its frequency and time resolution problem. In [20] [21], wavelet-based feature extraction technique to extract music features.

2.1.2 Machine Learning Approach

Once the features have been extracted, it is then necessary to find an appropriate pattern recognition method for classification. Fortunately, there are a variety of existing machine learning and heuristic-based techniques that can be adapted to this task.

Based on the statistical pattern recognition classifiers employed in the music genre classification, automatic genre classification can be categorized into two categories: prescriptive approaches and emergent approaches [13]. We propose two novel classification approaches for automatic genre classification in this thesis, one belongs to prescriptive approach (will be described in section 2.2) and the other belongs to emergent approach (will be described in section 2.3).

Prescriptive Approach

Aucouturier and Pachet [13] defined the prescriptive approach as an automatic

process that involves a two-step process: frame-based feature extraction followed by supervised machine learning method.

Tzanetakis [19] cited a study indicating that humans are able to classify genre after hearing only 250 ms of a signal. The authors concluded from this that it should be possible to make classification systems that do not consider music form or structure. This implied that real-time analysis of genre could be easier to implement than thought.

The ideas were further developed in [14], where a fully functional system was described in details. The authors proposed to use features related to timbral texture, rhythmic content and pitch content to classify pieces, and the statistical values (such as the mean and the variance) of these features were then computed. Several types of statistical pattern recognition (SPR) classifiers are used to identify genre based on feature data. SPR classifiers attempt to estimate the probability density function for the feature vectors of each genre. The Gaussian Mixture Model (GMM) classifier and K-Nearest Neighbor (KNN) classifier were respectively trained to distinguish between twenty music genres and three speech genres by feeding them with feature sets of a number of representative samples of each genre.

Pye [22] used MFCCs as the feature vector. Two statistical classifiers, GMM and Tree-based Vector Quantization scheme, are used separately to classify music into six types of Blues, Easy Listening, Classical, Opera, Dance and Rock.

Grimaldi [23] built a system using a discrete wavelet transform to extract time and

frequency features, for a total of sixty-four time features and seventy-nine frequency features. This is a greater number of features than Tzanetakis and Cook [14] used, although few details were given about the specifics of these features. This work used an ensemble of binary classifiers to perform the classification operation with each trained on a pair of genres. The final classification is obtained through a vote of the classifiers. Tzanetakis, in contrast, used single classifiers that processed all features for all genres.

In [24], Pamalk et. al. employed a K-NN classifier, combined with some clustering algorithm to group the similar music frames, to perform classification on four music collections.

It is impossible to give an exhaustive comparison of these approaches as these approaches use different target taxonomies and different training sets. However, we can still get some interesting observations.

Tzanetakis [19] achieved 61% accuracy using 50 songs belonging to 10 genres.

Pye [22] reported 90% on a total set of 175 songs over 5 genres.

Grimaldi[23] achieved a success rate of 82%, although only four categories are used.

Several remarks can be made from the above statement. A common remark is that features selection is very important for the music genre classification. Indeed, once significant features are extracted, any classification scheme may be used and is powerful enough to distinguish one or some music genres from others. Another

remark is that some types of music have proven to be more difficult to classify than others. For example, ‘Classical’ and ‘Techno’ are easy to classify, while ‘Rock’ and ‘Pop’ are not. A possible explanation for this is that the global frequency distribution of ‘Classical’ and ‘Techno’ is very different from other music types, whereas many ‘Pop’ and ‘Rock’ music use the same instrumentation. In other word, there are some relationships between different music genres. However, all the current prescriptive methods treated each music genre individually and equally and tried to use one classifier and unified features to classify music into different genres at one time. Little has been done to exploit the relationships among the music genres. The limitation of current prescriptive genre classification method exists in the fact that, the use of the unified feature set and classifier to classify the entire music genre database will not optimize the classification results. We will address this problem in our proposed hierarchical music genre classification.

Emergent Approach

In contrast to prescriptive approach, which assumes that genre taxonomy is given a priori, emergent approach, as its name indicates, tries to emerge a classification from the music database, by clustering songs according to a given measure of similarity. As we mentioned previously, there are two challenges in the prescriptive method: how to determine features to characterize the music and how to find an appropriate pattern recognition method to perform classifications. The more fundamental problem, however, is to determine the structure of the taxonomy in which music pieces will be classified. Unfortunately, this is not a trivial problem.

Different people may classify the same piece differently. They may also select genres from entirely different domains or emphasize different features. There is often an overlap between different genres, and the boundaries of each genre are not clearly defined. In [27], the authors perform genre classification experiment on manual labeling by human listeners, and from the human classification results, they got a conclusion that genre classification is inherently subjective and assumption that the consistent music taxonomy is given a priori is very weak. Therefore, the lack of universally agreed upon definitions of genres and relationships between them makes it difficult to find appropriate taxonomies for automatic classification systems, which prevents the perfect classification results to be expected from supervised learning methods.

In [25], Pachet and Cazaly attempted to solve this problem. They observed that the taxonomies currently used by the music industry were inconsistent and therefore inappropriate for the purpose of developing a global music database. They suggested building an entirely new classification system. They emphasized the goals of producing a taxonomy that was objective, consistent, and independent from other Metadata descriptors and that supported searches by similarity. They suggested a tree-based system organized based on genealogical relationships as an implementation, where only leaves would contain music examples. Each node would contain its parent genre and the differences between its own genre and that of its parent. Although merits exist, the proposed solution has problems of its own. To begin with, defining an objective classification system is easy, and getting everyone to agree on a

standardized system would be a far from easy task, especially when it is considered that new genres are constantly emerging. Furthermore, this system did not solve the problem of fuzzy boundaries between genres, nor did it deal with the problem of multiple parents that could compromise the tree structure.

Since there exist no good solutions for the ambiguity problem and due to inconsistencies in music genre definition, Pachet [26] presented the emergent approach as the best approach towards automatic genre classification. Rather than using existing taxonomies as in prescriptive systems, emergent systems attempted to emerge classifications according to certain measure of similarity. The authors suggested some similarity measurements based on audio signals as well as on cultural similarity gleaned from the application of data mining techniques to text documents. They proposed the use of both collaborative filtering to search for similarities in the text profiles of different individuals and co-occurrence analysis on the play lists of different radio programs and track listings CD compilation albums. Although this emergent system has not been successfully applied to raw music signals, the idea of automatically exploiting text documents to generate genre profiles is an interesting one.

So far, all the current music genre classification methods are supervised. The disadvantage is obvious: They are constrained by a fixed taxonomy, which suffer from ambiguities and inconsistencies as it has been described previously. In addition, to classify music genres, generally a large number of training examples for each genre must be collected and labeled. This is a labor-intensive and error-prone process which

is only feasible for a limited set of genres. Therefore, unsupervised music genre classification method needs to be investigated.

In the following two sections, we will present two contributions that we made to the area of the music genre classification, both in the machine learning stage. To be specific, in section 2.2, we propose a mutli-layer classifier based on SVM to discriminate music genres, which belongs to prescriptive approach. In this approach, the music classification problem can be solved by multi-layer classification scheme, in which the classifiers in different layer perform just two-class classifier and features used in each classifier are level dependent and genre specific features. The advantage of this method is that each classifier in hierarchical classification deals with an easier separable problem and we can use an independently optimized feature set at each step. In section 2.3, we propose an unsupervised music genre classification method, to avoid the ambiguities and inconsistencies caused by contrived taxonomy given a priori. Our proposed unsupervised classification approach takes the advantage of the similarity measure to organize the music collection with clusters of similar songs.

2.2 Hierarchical Music Genre Classification

To achieve good classification accuracy, we propose a multi-layer classifier based on SVM to discriminate musical genres. In the first layer, music is classified into Pop/Classical and Rock/Jazz music according to the features of beat spectrum and LPC-derived Cepstrum coefficients (LPCCs). In the second layer, Pop/Classical music is further classified into Pop and Classical music according to the features of

LPCCs and MFCCs, and Rock/Jazz music is further classified into Rock and Jazz music according to the features of zero crossing rates and MFCCs. SVM is used in all layers and each layer has different parameters and support vectors. The system diagram of hierarchical musical genre classification is illustrated in Figure 2-1.

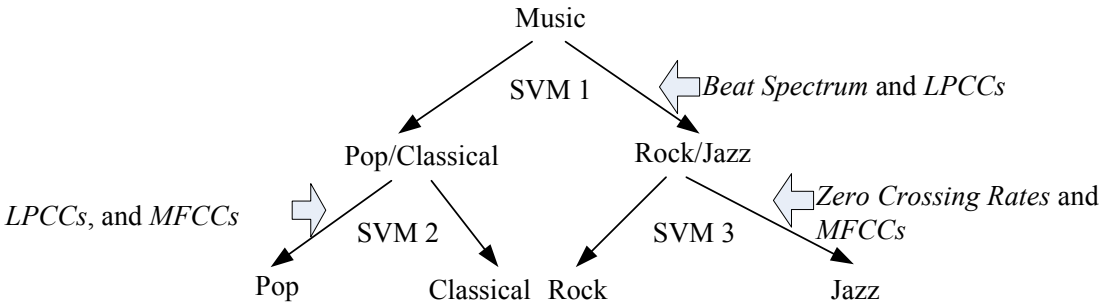


Figure 2-1: Music genre classification diagram

2.2.1 Feature Selection

Feature selection is important for music content analysis. The selected features should reflect the significant characteristics of different kinds of music signals. In order to better discriminate different genres of music, we consider the features that are related to temporal, spectral and rhythm aspects. The selected features here are Beat Spectrum, LPCCs, Zero Crossing Rate, and MFCCs.

Beat Spectrum

Beat spectrum [28] is a measure to automatically characterize the rhythm and tempo of the music. The beat spectrum can be defined as a measure of self-similarity as a function of the lag. Highly structured or repetitive music will have strong beat spectrum peaks at the repetition times. This reveals both tempo and the relative strength of particular beats, and therefore can distinguish between different kinds of rhythms. The calculation of beat spectrum can be found in Appendix A [A.1].

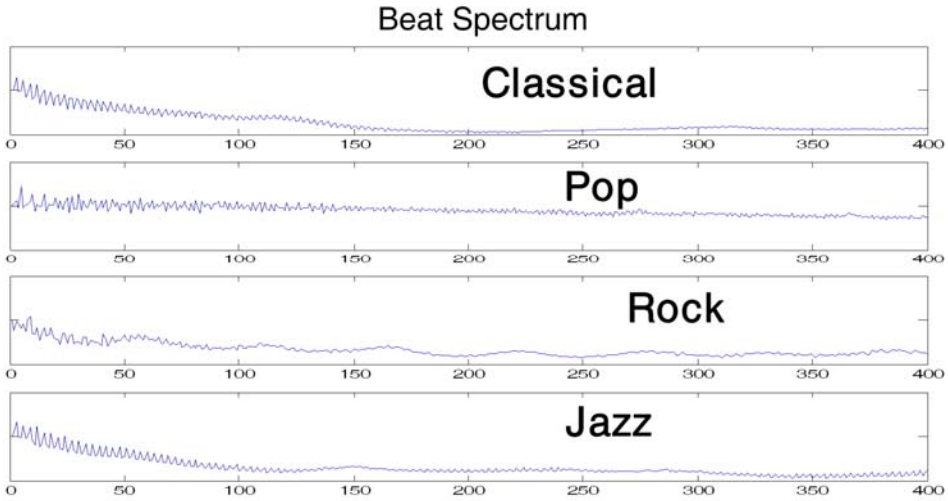


Figure 2-2: Beat spectrum for Classical, Pop, Rock and Jazz

Figure 2-2 illustrates the beat spectrum of Pop, Classical, Rock and Jazz music. The horizontal axis represents the time lag and the vertical axis represents the similarity magnitude. From the figure, we can see the different behavior of beat spectrum from four music genres.

LPC-derived Cepstrum coefficients (LPCCs)

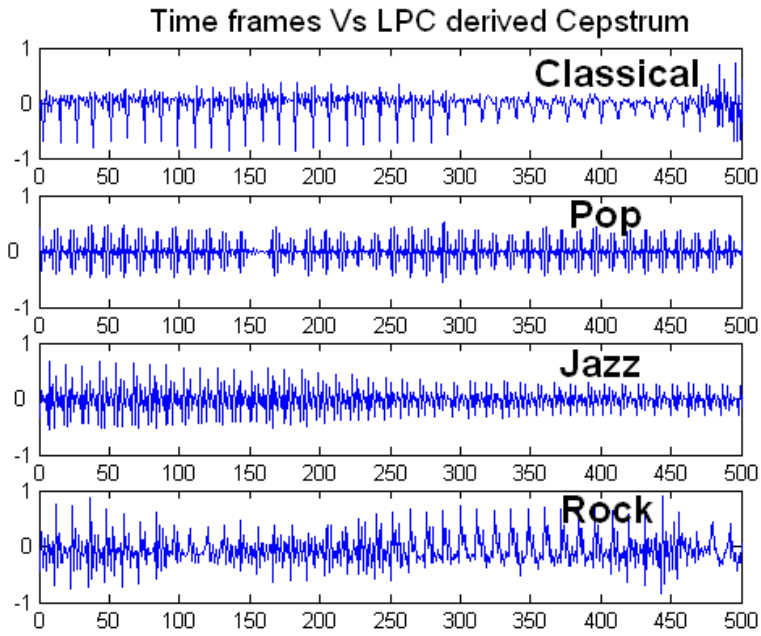


Figure 2-3: LPCCs for Classical, Pop, Jazz and Rock

The definition of LPCCs can be found in Appendix [A.2]. Figure 2-3 is an

example of LPCCs for the four music genres. The difference between the four music genres can be easily seen.

Zero Crossing Rates

The zero crossing is a useful feature in music analysis and the short-time zero crossing rate can be used to characterize music signal. The calculation of zero crossing rates can be found in appendix A [Appendix A.4].Figure 2-4 is an example of zero crossing rates for Rock and Jazz music. From the figure, we can see the characteristics of the zero crossing rates for Rock and Jazz are different.

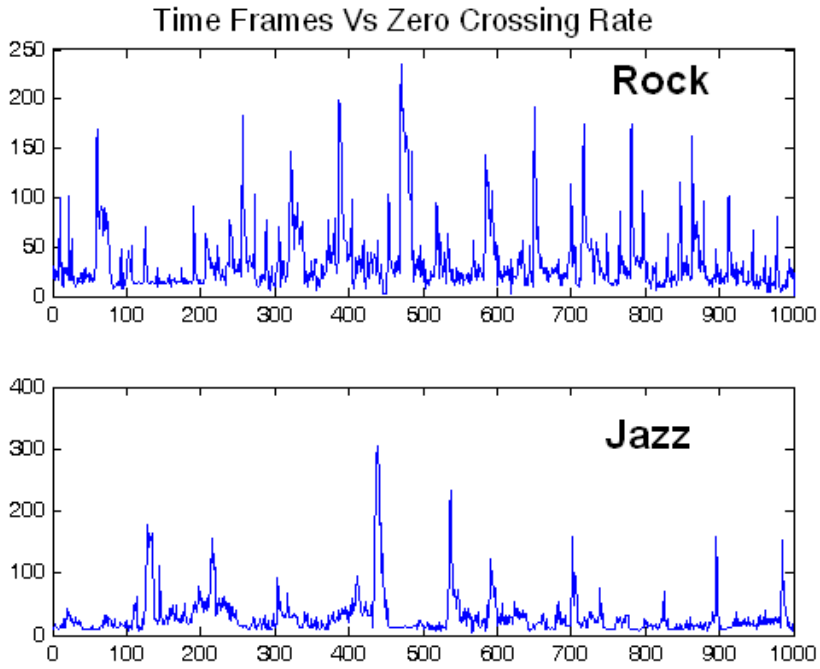


Figure 2-4: Zero crossing rates for Rock and Jazz music

Mel Frequency Cepstral Coefficients (MFCCs)

The mel-cepstral features can be illustrated by the Mel-Frequency Cepstral Coefficients (MFCCs) [Appendix A.5].Figure 2-5 is an example of 3rd MFCCs for Pop and Classical music. It can be seen that the variance is very high for the Pop music while it is considerably low for the Classical music.

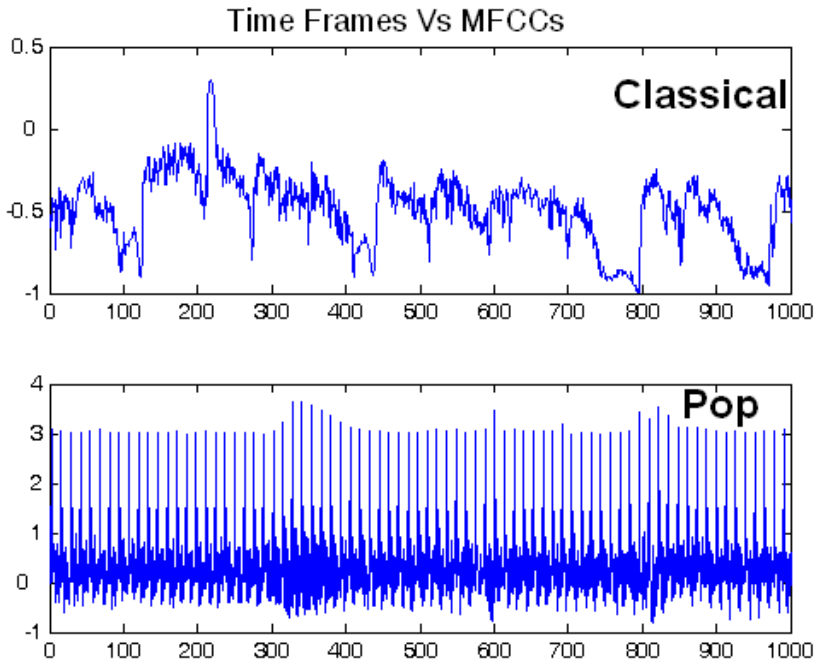


Figure 2-5: MFCCs for Pop and Classical music

2.2.2 Support Vector Machine (SVM) Learning

Support vector machine (SVM) learning is a useful statistical machine learning technique that has been successfully applied in the pattern recognition area [29][30]. We use non-linear support vector classifier to discriminate different musical genres. Therefore, classification parameters should be derived using support vector machine learning. The training process analyses musical training sample data to find an optimal way to classify musical frames into relevant genres. The derived classification parameters are used to discriminate different musical genres. Since we use three SVM classifiers and use different features to train these classifiers, the parameters corresponding to three classifiers are different.

The proposed hierarchical music genre classification approach has several merits. First, it solves the classification problem by using a number of SVM classifiers to

decompose it into a series of sub-problems, which enables the use of a divide-and-conquer approach and thus result in higher efficiency and accuracy. Second, it reduces the complexity of each single task. Third, it also improves the global accuracy by combining the results of the different SVM classifiers. Of course, the number needed classifiers is increased, yet, by having each of them handle a simpler problem, the overall required computational power is reduce. The experimental results show that the proposed hierarchical approach can get 92% classification accuracy. More results and details about the proposed hierarchical music genre classification can be referred to Chapter 5.

The main problem of our proposed hierarchical music genre classification approach is that the taxonomy and the classifier need to be maintained manually, which is a very expensive task, since the process requires domain experts to evaluate the relevance of music genres and find the optimal feature set for each classifier. In our proposed approach, we have to manually derive the music taxonomy and select the most suitable feature set for each classifier. The first level music taxonomy (music is classified into Pop/Classical and Rock/Jazz music) was obtained by comparison of the different classification result for the different combination of music genres with different feature set and choosing the best scheme. The features in the second level music taxonomy can be obtained by looking into the intrinsic relationship between the two music genres in the same categories and choosing the most discriminating features. Of course, manually building the music taxonomy in this manner is an expensive task. To solve this problem, Li and Ogihara [31] proposed an approach to

automatically infer genre relations from the confusion matrix generated from some efficient classifier, employing the linear discriminant projection. Another approach to address this problem is to automatically allow the emergence of a classification from the music database, by clustering songs according to a given measure of similarity of the songs in the database, which is described in the next section.

2.3 Unsupervised Music Genre Classification

To avoid the ambiguities and inconsistencies caused by fixed taxonomy given in the prescriptive classification approach, we also proposed an unsupervised music genre classification method which takes advantage of the similarity measure to organize the music collection without the taxonomy given a priori. Pachet [26] suggested using similarity measures based on cultural similarity to organize the music collections. This method differed with previous prescriptive method in emerging classifications according to some similarity measure. However, it works only for title and artist name appearing in the music sources, which is not always available in the music collections.

To the best of our knowledge, so far there is no unsupervised music genre classification method proposed based on low level music content, due to difficulty of measuring the similarity between the songs in the music database. To be specific, after the segmentation and feature extraction, each song is represented by a series of temporal vectors. How to measure the similarity of these time series from different songs is a problem. In our proposed approach, we address this problem by employing a Hidden Markov Model to model the relation between features over time.

Our proposed approach contains two steps. In the first step, as Figure 2-6 shows, every individual music piece is segmented into clips, and each clip is further segmented according to its intrinsic rhythmic structure. Features are extracted based on these segments. Considering the fact that, unlike most classical pattern recognition problems, the data we classified here are time series data. Therefore, we train a Hidden Markov Models (HMMs) to model the relationship between features over time. One good property of HMMs is that they provide a proper distance metric so that once each piece is characterized by its own HMM, and as a result, the distance between any pieces of the database can be computed. In the second step, we embed the distance between every pair of music pieces (HMMs) into a distance matrix and perform clustering to generate desired clusters.

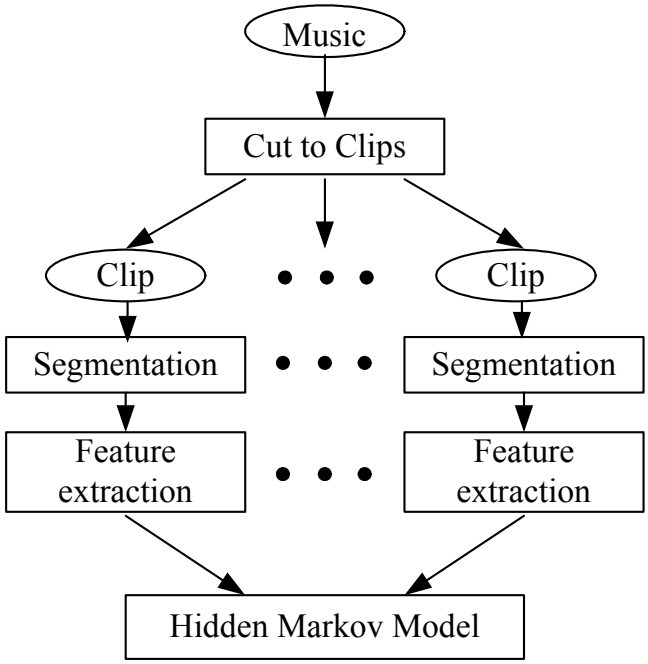
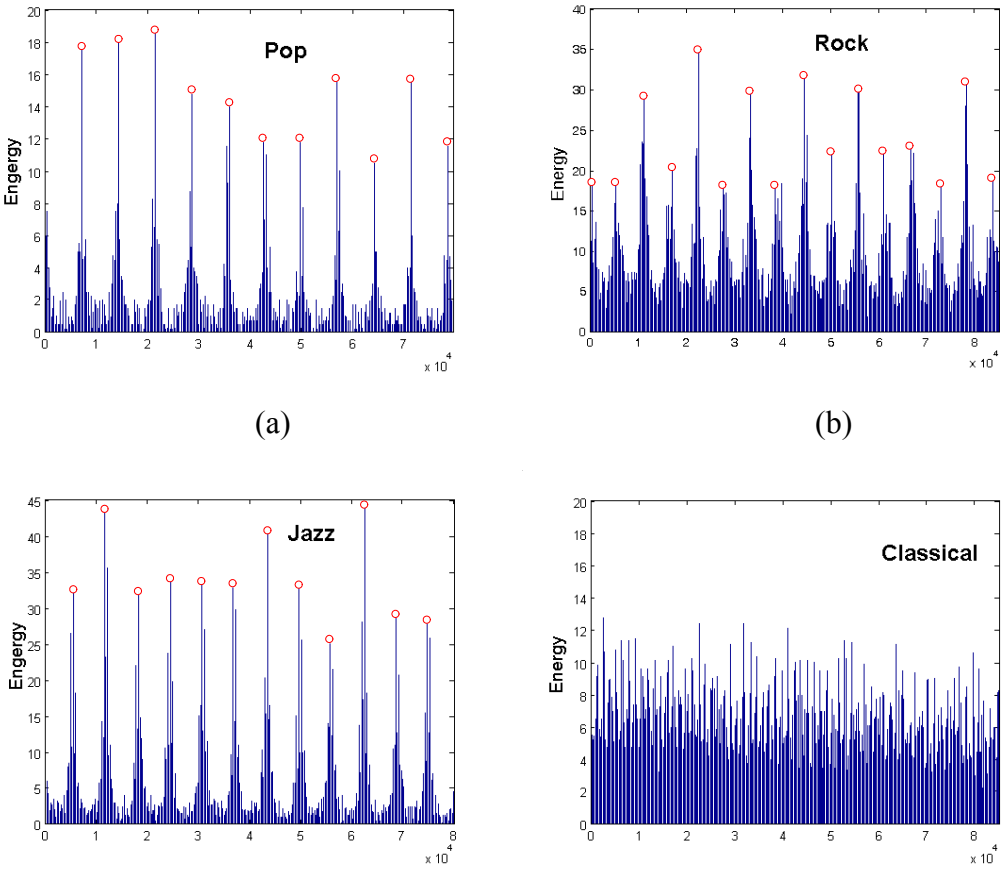


Figure 2-6: HMM training for individual music piece

2.3.1 Feature Selection

In order to better discriminate the different genres of music, we consider segmenting the music clip according to its intrinsic rhythmic structure. There are two reasons for using this segmentation scheme. Firstly, compared with the fixed length segmentation for music clips, segmenting music clips according to its intrinsic rhythm captures the natural structure of music genres better. Secondly, rhythmic structure characterizes the movement of music piece over time and contains such information as the regularity of the rhythm, beat, tempo, and time signature. These salient periodicities contain obvious time-sequential information which can be readily modeled by the HMMs. The different rhythmic structures for different genres are illustrated in Figure 2-7. The horizontal axis represents the sample index and the vertical axis represents the onset energy after autocorrelation. It can be seen that Pop, Rock and Jazz are highly structured music, and the inter-beat-interval, which is defined as the temporal difference between two successive beats, is almost a constant for a particular piece of music. However, the rhythmic structure varies for these three genres. As for Classical music, it is not a so highly structured music and the inter-beat-interval varies from time to time, which distinguishes it from other three genres.



(a) (b)

(c) (d)

Figure 2-7: Rhythmic structures for different genres

After the music clips have been segmented according to inter-beat-interval [32], three types of the features are extracted for each segment.

Mel-frequency cepstral coefficients (MFCCs)

The MFCCs feature was selected since it has been proven [14][22] to be efficient in music genre classification.

Linear prediction derived cepstrum coefficients (LPCCs)

The principal advantage of LPCCs is that they are generally decorrelated and this allows diagonal co-variances to be used in the HMMs.

Delta and acceleration

Delta Values [Δ ($\boldsymbol{V}_i$)] and acceleration values [acc ($\boldsymbol{V}_i$)] can be appended to any feature vector $\boldsymbol{V}_i$. They are computed as $\Delta(\boldsymbol{V}_i)= \boldsymbol{V}_i - \boldsymbol{V}_{i-1}$ and $acc(\boldsymbol{V}_i)= \Delta(\boldsymbol{V}_i) - \Delta(\boldsymbol{V}_{i-1})$, where $\boldsymbol{V}_i$ is a feature vector of either MFCCs or LPCCs.

Delta and acceleration values are very important improvements in feature extraction for HMMs because they effectively increase the state definition to include first and second order memory of past states.

2.3.2 Clustering by Hidden Markov Models

Our task is to classify observed low-level audio features into different music categories. Unlike most classic pattern classification problems, the data to be classified here are time series data, that is, a series of feature vectors. To handle this problem, Hidden Markov Models (HMMs) [33] is used. It can be completely defined by the number of hidden states, a static state transition probability distribution **A**, the observation symbol probability distribution **B** and the initial state distribution $\boldsymbol{\pi}$. We can define one HMM model as $\lambda= \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$.

Once the model topology and observation (training) vectors are determined, parameter estimation for the HMM is done using Baum-Welch algorithm [34].

As for clustering, an important issue is how to measure the similarity of music titles. HMMs provide a proper distance metric for sample comparison. The distance between two samples is defined as:

$$D(\mathbf{O}^{(1)},\mathbf{O}^{(2)})=\frac{\frac{1}{N_1}[\log P(\mathbf{O}^{(1)}\mid\lambda_1)-\log P(\mathbf{O}^{(2)}\mid\lambda_1)]}{2}+\frac{\frac{1}{N_2}[\log P(\mathbf{O}^{(2)}\mid\lambda_2)-\log P(\mathbf{O}^{(1)}\mid\lambda_2)]}{2} \quad (2\text{-}1)$$

where $\mathbf{O}^{(1)}=(o_1 o_2 \cdots o_{N_1})$ is a sequence of observations generated by HMM model λ_1 and $\mathbf{O}^{(2)}$ is generated by HMM model λ_2 . N_1 and N_2 are the length of $\mathbf{O}^{(1)}$ and $\mathbf{O}^{(2)}$ separately. The detail interpretations of Eq.(2-1) can be found in [Appendix B.2].

In our experiment, initially, we build a HMM model for each music piece. Considering the fact that a HMM model cannot be trained with only one sample (in our experiment, we found that the parameters of HMM model does not converge with one training sample.), we split one song into several clips, and each clip lasts for 30 second. These clips belonging to one music piece are used to train a HMM model. Assume there are N pieces of music in the database, then the distance between two music pieces can be calculated by Eq.(2-1) and the distance matrix **D** is $N\times N$ dimension. Given a distance matrix **D**, many clustering methods can be used. *K-means clustering* [35] is probably the simplest and most popular clustering algorithm. It allows portioning a set of vectors into K disjoint subsets. One of its weaknesse is that is requires the number of clusters (K) to be known in advance. However, in many real-world situations, the number of clusters is not known a priori. Therefore, we select *single-linkage hierarchical clustering* [35] algorithm as this method does bottom-up clustering. It starts with N singleton clusters and forms a sequence of clusters by successive merging. Of course, we can also use *complex-linkage hierarchical clustering* [35] to perform the clustering as these two hierarchical clustering methods do not make too much difference for well-separated

clusters. Here for the sake of simplicity, we only use *single-linkage hierarchical clustering*.

The experimental results show proposed music genre classification scheme is promising. The 5-hidden state HMM can achieve 89% average accuracy, which is comparable with the supervised genre classification proposed in section 2.2. More results and details about the proposed unsupervised music genre classification can be found in Chapter 5.

2.4 Summary

In this chapter, two novel music genre classification approaches were presented. At first, we proposed a hierarchy-based approach to discriminate music genres, which can be considered as an extension of current prescriptive approach for music genre classification. Then, in order to avoid the ambiguities and inconsistencies caused by fixed taxonomy given in the prescriptive classification approach, we also proposed an unsupervised classification approach to automatically emerge music genres from the database.

The main contributions include:

- A hierarchical classifier based on SVM was proposed to discriminate musical genres, which extend the current prescriptive approach for music genre classification not only reducing the complexity of each single task, but also improving the global classification accuracy.
- A segmentation scheme of the music signals based on music intrinsic

rhythmic structure analysis was proposed, which can better characterizes the movement of music piece over time and will be helpful in the process of subsequential HMM modeling.

- HMMs were employed to model the relationship between features over time from the raw songs. As a result, the similarity of each songs in music collections can be measured using the distance provided by the HMMs.

3 Music/Music Video Summarization

The aim of music summarization is to analyze the underlying structure of the individual songs in the database and finds the most salient part to represent the whole song. Music summarization is one of the two middle level applications developed in this thesis to structure the music database. It is important for music information retrieval especially for the presentation of the returned ranked list since it allows users to quickly hear the results of their query and make their selection. In automatic music summarization, machine learning is indispensable because not only finding the salient theme of a song needs semantic understanding by the computer, but also analyzing the underlying structure needs exploring the semantic regions with machine understanding. In this chapter, we present an approach to automatically summarize the song by first distinguishing the pure instrumental music and vocal music based on a machine learning approach, followed by an adaptive clustering algorithm on the selected vocal music segments to find the music structure. In addition, as an extension

of the music summarization problem, music video summarization will also be described in this chapter.

3.1 Related Work

Music summarization, as its name indicates, tries to analyze the underlying structure of individual music piece and finds the most salient part to represent the whole music. There are a number of techniques being proposed and developed to automatically generate summaries from text [36], speech [37]. Similar to text, speech summarization, music summarization refers to determining the most common and salient themes of a given music piece that may be used to represent the music and is readily recognizable by a listener. Automatic music summarization can be applied to music indexing, content-based music retrieval and web-based music distribution.

A summarization system for MIDI data has been developed [38]. However, MIDI format is not sampled audio data (i.e., actual audio sounds), instead, contains synthesizer instructions, or MIDI notes, to reproduce audio. Compared with actual audio sounds, MIDI data cannot provide a real playback experience and an unlimited sound palette for both instruments and sound effects. In this section, we focus on the music summarization for sound recording from real world.

Based on the methods employed to detect the repeating patterns, these approaches can be classified into two main categories:

Machine Learning Approaches

Machine learning approaches attempt to categorize each frame of a song into a

certain cluster based on the similarity distance between this frame and other frames in the same song. Then the number of frames in each cluster is used to measure the occurrence frequency. The final summary is generated based on the cluster that contains the largest number of frames. Since the music structure can be determined without prior knowledge, unsupervised learning is the natural choice. Clustering is the most widely used approach in this category, and several researchers have proposed various music structure analysis methods based on clustering.

The first real music summarization system was proposed by Logan & Chu [39]. They used clustering techniques to find the most salient part of a song, which is called the key phrase, in selections of popular music. They proposed a cross-entropy or Kullback Leibler (*K-L*) distance (See Appendix C.4 for detail description) to measure the similarity between different frames. Although merits exist, the proposed method had problems of its own. To begin with, K-L distance has the well-known disadvantages of slow convergence behavior and high computational cost. Furthermore, this system did not consider the music phrase boundaries problem, which will result in incomplete music phrases contained in the final summary. From the view point of listeners, these incomplete music phrases are not acceptable when they listen to the summary.

The ideas were further developed in [40] where a fully functional system was described in detail. The authors employed the Mahalanobis distance for similarity measure rather than the *K-L* distance when they clustered the frames, since Mahalanobis distance converges faster than K-L distance and has low computational

cost. However, the music phrase boundary problem was still not considered in this method. In addition, the fixed overlap ratio of music segmentation scheme will prevent the algorithm from optimally grouping the music frames.

Lu & Zhang [41] proposed to use two-pass approach to generate the music summary. In the first pass, they used a clustering method to group the frames. In the second pass, they used estimated phrase length and phrase boundary confidence of each frame to detect the phrase boundary. In this way, the final music summary would not include the broken music phrases. However, when performing summarization, this approach did not consider the different roles played by pure instrumental music and vocal singing in a song, (i.e. the most distinctive or representative music themes should repetitively occur in the vocal part of an entire music work). As a result, the final summary may contain some undesired pure instrumental music portions.

Pattern Matching Approaches

The pattern matching approach aims at matching the underlying excerpt with the whole song in order to find the most salient part. The best matching excerpt can be the one that is most similar to the whole song or the one that is repeated most often in the whole song.

Foote and Cooper [42][43] first introduced pattern matching approach to music summarization. They proposed a representation called similarity matrix for visualizing and analyzing the structure of music. One attempt of this representation was to locate points of significant change in music, which they called audio novelty. The audio novelty score is based on the similarity matrix, which compares frames of

music signals based on features extracted from the audio. The summary was one consecutive excerpt which was selected to maximize quantitative measures of the similarity between candidate excerpts and the source audio as a whole. Bartsch and Wakefield [44] used the similar pattern matching approach while they use different features, the chroma-based features, to represent the music content. However, the drawback of such kind of pattern matching approach is that the distance function used to measure similarity between different frames may fail to capture the similarity of the dynamic characteristic of the consecutive music frames. As a result, some 'false matching" excerpt would be selected while the optimal excerpt would not be selected as the summary.

Chai & Vercoe [45] proposed a dynamic programming method to detect the repetition of a fixed length excerpt in a song one by one. First, they segmented the music into frames, and grouped the fixed number of frames into excerpts. Then, they employed a dynamic programming method to measure the repetitive property of each excerpt in the song. The consecutive excerpts that had the same repetitive property were merged into sections and each section was labeled according to the repetitive relation (i.e., each section was given a symbol such as "A","B", etc). The final summary was generated based on the most frequently repeated music section. Although this method can identify the most repeated segments from a music piece, it still did not consider distinguishing between pure instrumental music and vocal music when generating the music summary.

The current methods for music summarization mostly focus on finding the most

salient part of a music piece. However, they all fail to consider distinguishing the pure instrumental music and vocal music during the process of music summary generation. As a result, a summarized segment may contain the undesired pure instrumental music portions. This is definitely not desirable for the purpose of understanding music content, since according to music theory, the most distinctive or representative music themes should repetitively occur in the vocal part of a music work.

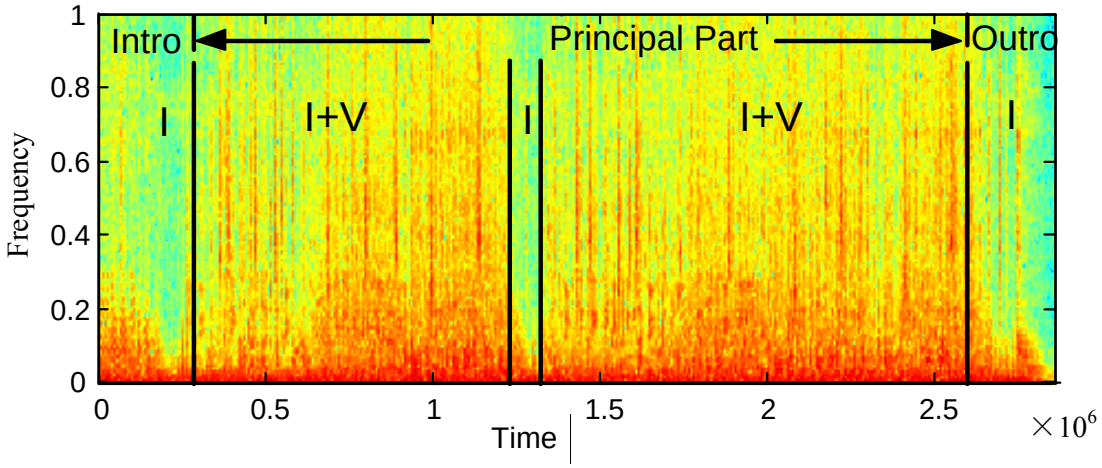
As an extension of music, Music Video (MV) is one video genre popular among music fans today. Nowadays, most MV summaries are manually produced. In contrast to other video genres, automatic video summarization has been applied to sports video [46][47], news video [48][49], home video [50][51] and movies [52]. Although recent work of video summarization techniques on music video has been reported [53][54], this work used high-level information such as titles, artists and closed captions other than low-level audio/visual features to generate the music video summary. However, such high-level metadata are not easily obtained directly from the music video content. Therefore, assumption of availability of such metadata makes the problem easier and is not feasible for automatic summarization based on music video content only. Our approach proposed in this thesis is to generate music video summary based on low-level audio/visual features which can be directly obtained from the music video content. To the best of our knowledge, there is no summarization technique available for music videos using low-level audio/visual features.

In our proposed music summarization approach, after the feature extraction, we first distinguish the pure instrumental music and vocal music based on a machine

learning approach, and then propose an adaptive clustering algorithm on the selected vocal music segments to find the main theme of a song.

3.2 The Proposed Music Summarization

In our proposed music summarization scheme, music structure analysis is important. We found that normally a music song ("Top of the world" by Carpenter) is composed of three parts: Intro, Principal and Outro, as shown in Figure 3-1. The vertical axis represents the normalized frequency and the horizontal axis represents the sample index. In Figure 3-1, 'V' represents the 'pure singing voice' and 'I' represents the 'pure instrumental music'. The combination, 'I+V', refers to 'vocal music' which is defined as the music containing both singing voice and instrumental music. The Intro and Outro parts usually contain pure instrumental music without vocal components while the principal part contains a mixture of the vocal and instrumental music as well as some pure music portions. The 'pure music' here is defined as the music that contains only instrumental music lasting for at least 3 seconds. This is because pure music used to bridge different parts is normally of more than 3 seconds duration, while the music between the music phrases within the verse or chorus is of less than 3 seconds. Thus, it cannot be treated as the pure music. Because these three parts play different roles in conveying music information to listeners, we treat them separately when creating the music summary (See Section 3.2.2 for a detailed description).



I: Instrumental Music
I+V:Vocal and Instrumental Music

Figure 3-1: Typical music structure embedded in the spectrogram

For each part of the music, the content is segmented into fixed-length and overlapping frames. Feature extraction is performed in each frame. Based on the calculated features, an adaptive clustering algorithm is applied to group these frames to obtain the structure of the music content. Finally, the music summary is created based on the clustered results and music domain knowledge.

3.2.1 Feature Extraction

Feature extraction is very important for music content analysis. The extracted features should reflect the significant characteristics of the music content. Commonly extracted features include Linear Prediction Coefficient derived Cepstrum coefficients (LPCCs), Zero-Crossing Rates (ZCR) and Mel Frequency Cepstral Coefficients (MFCCs).

Linear Prediction Coefficients (LPCs) and LPC derived Cepstrum coefficients (LPCCs)

Linear prediction and linear prediction derived cepstrum are two linear prediction

analysis methods [12] and they are highly correlated to each other. The basic idea behind the linear predictive analysis is that a music sample can be approximated as a linear combination of past music samples. By minimizing the sum of the squared differences (over a finite interval) between the actual music samples and the linear predictive ones, a unique set of predictor coefficients can be determined. The calculation of LPCs and LPCCs can be found in appendix [Appendix A.2 and A.3]. Experiment shows that LPCCs is much better than LPCs in identifying the vocal music [55].

Generally speaking, the performance of LPCs and LPCCs can be improved by (20~25) % by filtering the full band music signal (0 ~ 22.05 kHz with 44.1 kHz sampling rate) into sub-frequency bands and then down-sampling the sub-bands before calculating the coefficients.

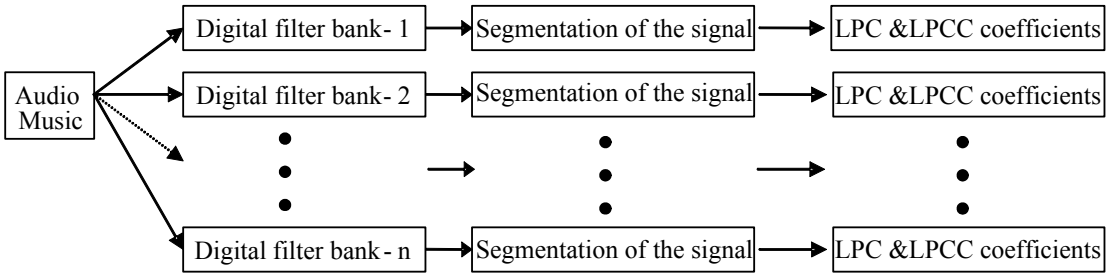


Figure 3-2: Block diagram for calculating LPCs & LPCCs

The sub-bands are defined according to the lower, middle and higher music scales [56], as shown in Figure 3-2. Frequency ranges for the designed filter banks are *[0-220.5], [220.5-441], [441-661.5], [661.5-882], [882-1103], [1103-2205], [2205-4410], [4410-8820], [8810-17640],* and *[17640-22050] Hz.* Therefore calculating LPCs for different frequency bands can represent the dynamic behavior of the spectrums of the selective frequency bands (i.e. different octave of the music).

Zero-crossing rates (ZCR)

In the context of discrete-time signals, a zero-crossing refers to two successive samples having different algebraic signs. The rate at which zero-crossings occur is a simple measure of the frequency content of a signal. This average zero-crossing rate gives a reasonable way to estimate the frequency content of a signal. While ZCR values of instrumental music are normally within a relatively small range, the vocal music is often indicated by high amplitude ZCR peaks resulted from pronunciations of consonants [57]. Therefore, ZCR values are useful for distinguishing vocal and pure music.

Figure 3-3 is an example of zero-crossing rates for the vocal music and pure music. It can be seen that the vocal music has higher zero-crossing rates than pure music. This feature is also quite sensitive to vocals and percussion instruments. Mean values are 188.247 and 47.023 for vocal music and pure music respectively.

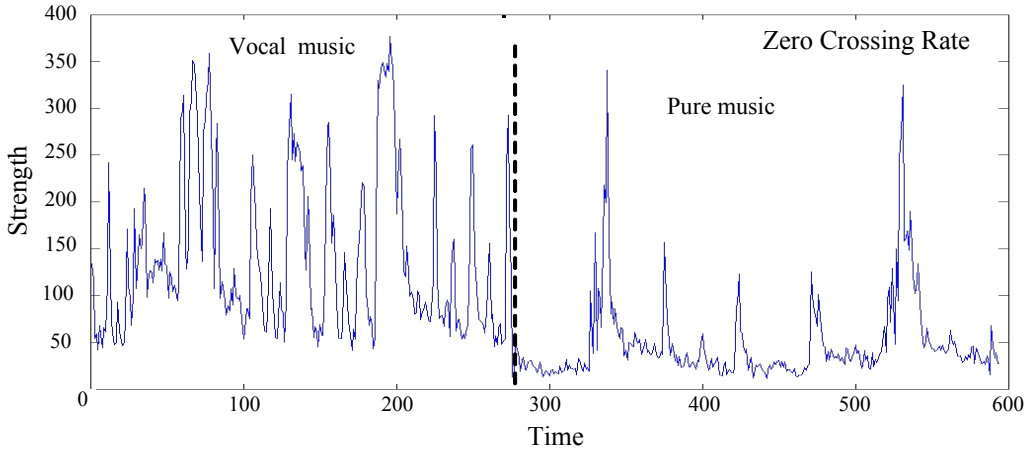


Figure 3-3: Zero-crossing rates (0-276 second is vocal music and 276-592 second is pure music)

Mel-Frequency Cepstral Coefficients (MFCCs)

The mel-cepstral features have proven to be highly effective in automatic speech

recognition and in modeling the subjective pitch and frequency content of the audio signals.

MFCCs are good features for analyzing the music because of the significant spectral differences between human vocalization and musical instruments [58]. Figure 3-4 is an example of MFCCs for vocal music and instrumental music. It can be seen that the mean value is 1.3704 for the vocal music and 0.9288 for pure instrumental music. The variance is very high for the vocal music while it is considerably low for the pure music.

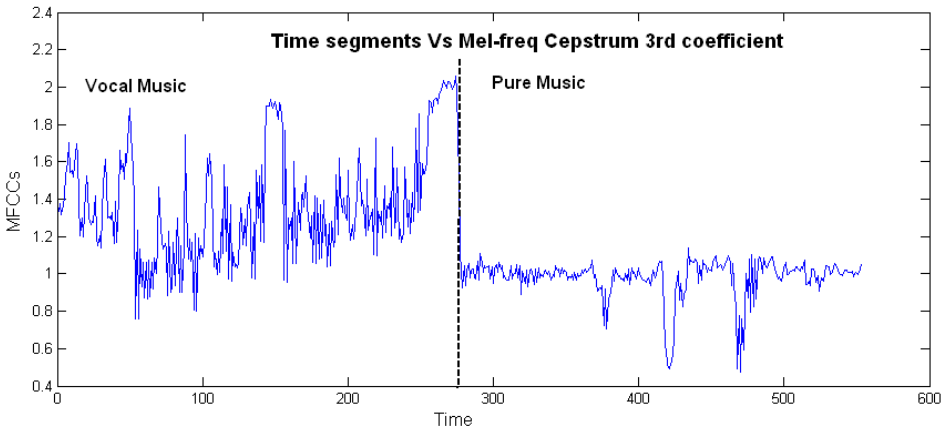


Figure 3-4: The 3rd MFCCs (0-276s is vocal music and 276-573s is pure music)

3.2.2 Music Classification

The purpose of music classification is to analyze a given music sequence to identify the pure music and the vocal music segments. According to the music theory, the most distinctive or representative music themes should repetitively occur in the vocal part of an entire music work [59] and the summary should focus on the mixture portion (The instrumental-only music is not considered in this thesis). Therefore the pure

music in the principal part is not the key component of a song (mostly the pure music in the principal part is the bridge between the chorus and verse) and can be discarded. But for the pure music in the Intro and Outro, it contains information indicating the beginning and the end of the music work and cannot be ignored. Therefore, if the pure music segment is detected at the beginning and the end of the music sequence, it will be identified as the Intro and Outro part, separately. We will retain these two parts in the music summary. For the pure music in the principal part, we discard it and create only a summary of mixed music in principal part.

Based on calculated features (LPCCs, ZCR and MFCCs) of each frame, we employ a non-linear support vector classifier to discriminate the vocal and pure music. The Support Vector Machine (SVM) technique is a useful statistical machine learning technique that has been successfully applied in the pattern recognition area [29][30]. Figure 3-5 illustrates a conceptual block diagram of the training process to produce classification parameters of the classifier.

The training process analyses music training data to find an optimal way to classify music frames into pure or vocal class. The training data are segmented into fixed-length and overlapping frames (in our experiment we used 20ms frames with a 50% overlapping). Features such as LPCCs, ZCR and MFCCs are calculated from each frame. The SVM methodology is applied to produce the classification parameters according to the calculated features. The training process needs to be performed only once. The derived classification parameters are used to classify frames as pure and vocal music.

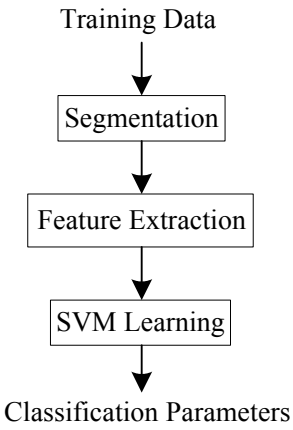


Figure 3-5: Diagram of the SVM training process

After training, the derived classification parameters are used to identify the pure music and vocal music. For a given music track:

1) Segment it into fixed-length frames;
2) For every frame, extract features such as LPCCs, ZCR and MFCCs to construct the feature vector;
3) Input each feature vector to a trained SVM and the SVM will label the corresponding frame as the pure music or vocal music;
4) For those “pure music” frames labeled, if the continuous frames last for more than 3 seconds, identify them as a pure music portion.

For the pure music portion located at the head and tail of a music piece, we retain them for the next processing step, while the other pure music portions are discarded.

3.2.3 Clustering

All methods mentioned above use the fixed overlap rate segmentation scheme to segment the music frames. However, in the initial stage, it is difficult to exactly determine the proper length of the overlap. As a result, the fixed overlapping rate

segmentation cannot guarantee ideal results of the frame grouping. In our proposed method, based on the calculated features of each frame, we use an adaptive clustering method to group the music frames and obtain the structure of the music. Two of the issues associated with the music segmentation are the length and the degree of overlap of the segmentation window. An inappropriate choice of these two will affect the final clustering result. For speech signals, a typical segmentation window size is 20ms, as the speech signal is generally treated as being stationary over such time intervals. Considering popular music, the tempo of a Pop song is constrained between 30-150 M.M (Mälzel's Metronome: the number of quarter notes per minute) and is almost constant [60], and the signals between two notes can be thought as being stationary. Therefore, the time interval between two quarter notes can range from 400ms to 2000ms (The time interval for Quaver and Semiquaver are multiple of the time interval of quarter notes). We choose the smaller one as our segmentation window size. As we mentioned, the overlapping length of adjacent frames is another issue associated with the music segmentation. If the overlapping length is too long, the redundancy of two adjacent frames will be high, and on the other hand, if the overlapping length is too short, the time resolution of the signals will be low. In the initial stage, it is difficult to exactly determine the proper length of the overlapping. But we can adaptively adjust the overlapping length if the clustering result is not ideal for frame grouping. This is the key point in our algorithm which differs from the non-adaptive clustering algorithm proposed in [39].

The clustering algorithm is described as follows.

1) Segment the music signal (vocal or pure music) into w fixed-length (w is 400ms in this case) and λ_p% overlapping frames and label each frame with a number i (i=1, ⋯, n), where overlapping rate λ_p =*10*p, (p =1, 2, 3, 4, 5, 6)*. Here we vary λ_p at a step of 10 (empirically derived) because a smaller step (i.e. 1 or 2) will make our algorithm computationally expensive.

2) For each frame, calculate the music features to form a feature vector:

$$\vec{V}_i = (LPCC_i, ZCR_i, MFCC_i) \quad i = 1, 2, \ldots, n \tag{3-1}$$

3) Calculate the distances between every pair of the music frames i and j using the Mahalanobis distance [61]:

$$D_M(\vec{V}_i, \vec{V}_j) = [\vec{V}_i - \vec{V}_j] R^{-1} [\vec{V}_i - \vec{V}_j] \qquad i \neq j \tag{3-2}$$

where $\boldsymbol{R}$ is the covariance matrix of the feature vector. The reason we use Mahalanobis distance is that it is very sensitive to inter-variable changes in all dimensions of the data.

Since $\boldsymbol{R}^{-1}$ is symmetric, it is a semi or positive matrix. It can be diagonalized as $\boldsymbol{R}^{-1}=\boldsymbol{P}^T\boldsymbol{\Lambda}\boldsymbol{P}$, where $\boldsymbol{\Lambda}$ is a diagonal matrix and $\boldsymbol{P}$ is an orthogonal matrix. Thus Equation (3-2) can be simplified in terms of Euclidean distance as follows:

$$D_M(\vec{V}_i, \vec{V}_j) = D_E(\sqrt{\Lambda} P\vec{V}_i, \sqrt{\Lambda} P\vec{V}_j) \tag{3-3}$$

Since $\boldsymbol{\Lambda}$ and $\boldsymbol{P}$ can be computed directly from $\boldsymbol{R}^{-1}$, the computational complexity of the vector distance can be reduced from O (n^2) to O (n).

4) Embed the calculated distances into a two-dimensional matrix $\boldsymbol{\Psi}$ which

contains the similarity metric calculated for all frame combinations, hence frame indexes *i* and *j* such that the $(i, j)^{th}$ element of $\boldsymbol{\Psi}$ is $D(i, j)$.

5) Normalize matrix $\boldsymbol{\Psi}$ according to the highest distance between frames. i.e. $0 \le D(i, j) \le 1$.

6) For a given overlapping rate λ_p, calculate the sum total of distances between all frames, denoted as S_d , which is defined as follows:

$$S_d = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} D(i, j) \tag{3-4}$$

7) Repeat steps 1) – 6) by varying the overlapping rate λ_p, an optimal λ_p^* can be found which can give the maximum value for S_d. In our experiments, we found that about 80% of songs have the optimal λ_p^* =30, about 18% of songs have the optimal λ_p^* =20 and 40, and less than 2% of the songs have the optimal λ_p^* taking the other values, i.e. 10, 50, 60.

8) Do Agglomerative Hierarchical Clustering [35].

Here we consider putting *n* music frames into C^* clusters. At initial stage, we start with *n* singleton clusters and form C^* clusters by successive merging using a bottom-up manner. Here, C^* is the optimal desired number of clusters which can be defined as follows:

$$C^* = k \cdot \left\lceil \frac{L_{sum}}{T_c^*} \right\rceil \tag{3-5}$$

where L_{sum} is the time length of the music summary (in seconds) and T_c^* is the minimum time length of the sub-summary generated in a cluster (for sub-summary

generation, see Section 3.2.4 for details). Factor k is a scaling constant selected in the experiment and it is better to select the number of clusters k times more than the required number of clusters to guarantee enough clusters to be selected in the summary. Our human study experiment has shown that the ideal time length of a sub-summary is between 3 and 5 seconds. A playback time which is shorter than 3 seconds will result in a non-smooth and has non-acceptable music quality, while a playback time which is longer than 5 seconds will result in a lengthy and slow-paced one. Thus, $T_c^* = 3$ has been selected for our experiment.

The detailed procedure for Agglomerative Hierarchical Clustering can be described as follows:

Procedure

1) Let $C = n$, $\vec{V}_i \in H_i$, $i = 1,\ldots,n$, where C is the initial number of clusters and H_i denotes the i^{th} cluster. Initially, one cluster contains one frame.
2) If $C=C^*$, stop. C^* is the desired number of clusters.
3) Find the "nearest" pair of distinct clusters, H_i and H_j, where i and j are cluster indexes.
4) Merge H_i and H_j, delete H_j, and $C \leftarrow C$-1.
5) Go to step 2).

At any level, the distance between the nearest clusters can be used as dissimilarity values for that level. Dissimilarity measures can be calculated by

$$d_{mean}(H_i, H_j) = \left\| m_i - m_j \right\| \tag{3-6}$$

where m_i and m_j are mean values of all vectors belonging to the cluster H_i and H_j.

3.2.4 Summary Generation

After clustering, the structure of the music content can be obtained. Each cluster contains frames with similar features. The summary can be generated in terms of this structure and domain-specific music knowledge.

According to music theory, the most distinctive or representative music themes should repetitively occur over the duration of the entire piece [59]. Based on this fact and clustering results, the summary of a music piece can be generated as follows:

Assume the summary length is $1000 \cdot L_{sum}$ microsecond (ms); the number of clusters is C^*; the music frame length is w ms.

1) The total number of music frames in the summary can be calculated as:

$$n_{total} = \frac{1000 \cdot L_{sum} - w \cdot \lambda_p\%}{(1-\lambda_p\%) \cdot w} \tag{3-7}$$

where λ_p is the overlapping rate defined in Section 3.2.3. The equation can be derived from the fact that the final summary (with the length of $1000 \cdot L_{sum}$ ms) is padded by n_{total} overlapped music frames with w ms frame length and λ_p% overlapping rate.

2) According to the cluster mean distance matrix, we arrange the distance between cluster pairs in descending order and the higher distance clusters are selected for generating the summary for the purpose of maximizing the coverage of music contents in the final summary.

3) Sub-summaries are generated within the cluster. Selected frames in the cluster must be as continuous as possible and the length of the

combined frames within the cluster should be 3s~5s or the number of frames should be between n_s frames and n_e frames, where:

$$n_s = \frac{3000 - w \cdot \lambda_p\%}{(1 - \lambda_p\%) \cdot w} \tag{3-8}$$

and

$$n_e = \frac{5000 - w \cdot \lambda_p\%}{(1 - \lambda_p\%) \cdot w} \tag{3-9}$$

Assume F_i and F_j are the first frame and last frame in the time domain of a selected cluster such that ($j > i$) and $n_c = (j\text{-}i) > 1$.

From music theory and our user study experiment, a piece of music with discontinuous frames is not acceptable to human ears. Based on this, we should generate the continuous sub-summaries. If frames are discontinuous between frame F_i and frame F_j, we first add frames between F_i and F_j, make the frames in this cluster continuous, and at same time delete these added frames from other clusters; we then follow the condition (1), (2), or (3) to adjust the sub-summary length within the cluster to meet the sub-summary length requirement defined in Equation (3-8) and Equation (3-9).

Condition (1): $n_c < n_s$, as Figure 3-6(a) shows, we add frames before the head (F_i) and after the tail (F_j) until the sub-summary length is equal to n_s.

Assume x represents the required number of added frames before F_i (head frame), and y represents the required number of the added frames after F_j (tail frame). Initially, x should be close to y, which means the added frames before F_i and after F_j are distributed in a balanced manner. Therefore, x and y can be calculated

as:

$$x = \lfloor (n_s - n_c)/2 \rfloor \tag{3-10}$$

$$y = n_s - x \tag{3-11}$$

However, if added frames exceed the first frame or the last frame of the original music, exceeding frames will be added to the tail or the head, respectively. After adjusting, the actual number of the added frames before F_i and after F_j , denoted as x' and y' respectively, can be calculated as following:

$$x' = i - 1; \quad y' = y + (x - x') \tag{3-12}$$

$$y' = (n - j) + 1; \quad x' = x + (y - y') \tag{3-13}$$

where n is the total number of frames in the music.

Equation (3-12) calculates the actual number of the added frames before F_i and after F_j , when the required number of added frames before head frame F_i exceeds the first frame of the original music. The actual number of the added frames before F_i is (i-1) and the rest frames of x will be added to the tail. Therefore, the actual number of the added frames after F_j is y + (x-x'). A similar analysis can also be applied to Equation (3-13), which calculates the actual number of the added frames before F_i and after F_j , when the required number of added frames after the tail frame F_j exceeds the last frame of the original music.

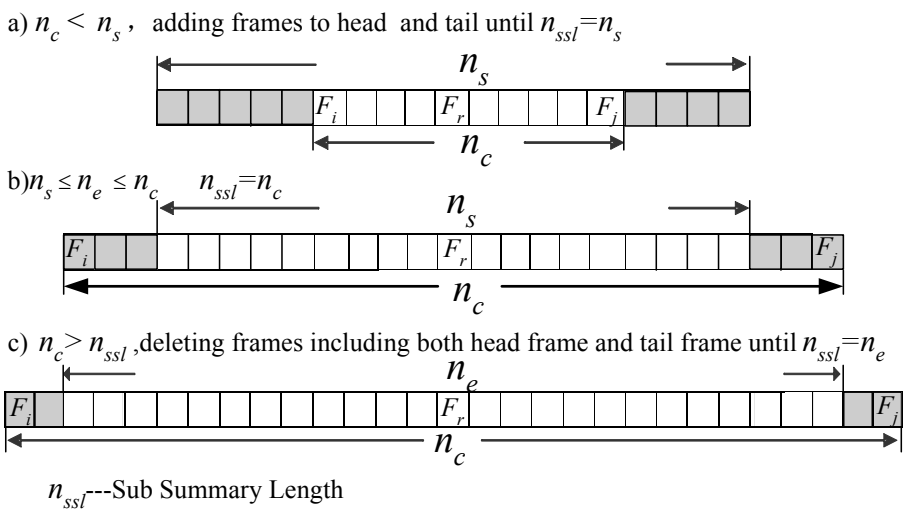


Figure 3-6: Sub-summaries generation

Condition (2): $n_s \le n_c \le n_e$, as Figure 3-6(b) shows, no change is made in the sub-summary length and it is equal to n_c.

Condition (3): ($n_c > n_e$), as Figure 3-6(c) shows, we delete frames both from the head frame and the tail frame until the sub-summary length is equal to n_e.

4) Repeat step 3) to generate individual sub-summaries for another selected cluster and stop the process when the summation of the sub-summary length is equal to or slightly greater than the required summary length.

5) If the summation of the sub-summary length exceeds required summary length, we find the last sub-summary added to the music summary and adjust its length to fit the final summary length.

6) Merge those sub-summaries according to their positions in the original music to generate the final summary.

3.3 Music Video Summarization

After the music track has been summarized, the generated music summary can be used as the basis of the music video summary. In some sense, music video summarization can be considered as an extension of music summarization. Nowadays, many music companies are putting the music videos on websites, and customers can purchase them via the Internet. However, from the customer point of view, they would prefer to watch the highlights before making their purchases. On the other hand, from the music company point of view, they would be glad to provoke the buying interests of the music fans by showing the highlight of a music video rather than showing everything, as there are no profits for company if they allow the music fans download the whole music video freely.

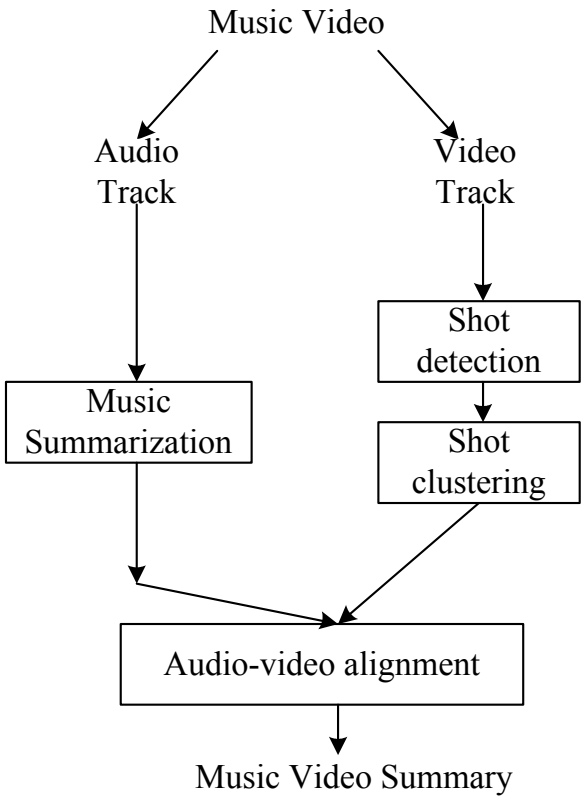


Figure 3-7: Block diagram of proposed summarization system

Music video summaries are available on some music websites, but they are generated manually, which is very labor intensive and time-consuming. Therefore, it is crucial to come up with an automatic summarization approach for music videos. Figure 3-7 is the block diagram of the proposed approach. As figure shows, the music video is separated into the music track and the video track. For the music track, a music summary is created by employing the music summarization scheme described previously. For the video track, shots are detected and clustered using visual content analysis. Finally, the music video summary is created by specially aligning the music summary and clustered visual shots.

3.3.1 Music Video Structure

Video programs such as movies, dramas, talk shows, etc, have a strong synchronization between their audio and visual contents. Usually what we hear from the audio track directly explains what we see on the screen, and vice versa. For this type of video program, since synchronization between audio and image is critical, the summarization strategy has to be either audio-centric or image-centric. The audio-centric summarization can be accomplished by first selecting important audio segments of the original video based on certain criteria and then concatenating them together to compose an audio summary. To enforce the synchronization, the visual summary has to be generated by selecting the image segments corresponding to those audio segments in the audio summary.

Similarly, an image-centric summary can be created by selecting representative

image segments from the original video to generate a visual summary, and then taking the corresponding audio segments to generate the associated audio summary. For both summarization approaches, either audio or visual content of the original video will be sacrificed in the summaries.

However, the music video is a special type of video. The visual and audio content combination in the music video can be divided into two categories: the polyphonic structure and homophonic structure [62]. In a polyphonic structure, the visual content does not in any way parallel the lyrics of the music. The visual content seems to tell its own story and is relatively independent of the meaning of the lyrics. For example, while the music proclaims the tender love, the pictures may show surprisingly violent scenes. For these music videos, due to their weak synchronization between the visual and audio content, summarizing the visual and audio track separately and then sticking them together appears to be satisfactory.

In a homophonic structure, the lyrics of the music, or at least its major literal themes, are in step with the visual event with similar meanings. According to [62], the picture and sound in these videos are organized as an aesthetic whole using some matching criteria such as historical matching, geographical matching, thematic matching and structure matching, etc. For the music videos in this category, on the one hand, we can summarize them using the same method as audio-centric and image-centric summarization, which enforces the synchronization but has to sacrifice either audio or visual content of the original video. On the other hand, we can also use the same summarization approach as the polyphonic structure music video, which

enforces the maximum coverage both for video and audio content but has to sacrifice synchronization thereof. In other words, we have to trade off between the maximum coverage and synchronization.

Considering the human perception, there is an asymmetrical effect of audio-visual temporal asynchrony on the auditory attention and visual attention [63]. The auditory attention is sensitive to audio-visual asynchrony while the visual attention is insensitive to the audio-visual asynchrony. Therefore, the minor deviation for the visual content from the music is allowed in the range of human perceptual acceptance. Based on above analysis, we use the same summarization approach for the music video in homophonic structure as the one used in polyphonic structure, which can maximize the coverage for both audio and visual contents without having to sacrifice either one of them, at the cost of some potential asynchrony between the audio and video track.

However, we have realized that the ideal summarization scheme for music video in homophonic structure should have the maximum coverage and strict synchronization for the visual and auditory content. This can be achieved by semantic structure analysis both for the visual and music content and will be addressed in the future work.

3.3.2 Shot Detection and Clustering

To summarize the visual content of the music video, we need to turn the raw video sequence into a structured data set $\boldsymbol{W}$ (named as clustered shot set here), where

boundaries of all camera shots are identified and visually similar shots are grouped together.

In the clustered shot set $\boldsymbol{W}$, any pair of the clusters in $\boldsymbol{W}$ must be visually different, and all the shots belonging to the same cluster must be visually similar. The total number of clusters varies depending on the internal structure of the original video.

It has been shown [50] that video programs with more than one shot cluster where each has an equal time length will have the minimum redundancy. It has been also mentioned that for the purpose of reviewing the visual content, the ideal playback length for each shot cluster is between 1.5 to 2.0 seconds [50]. A playback time which is shorter than 1.5 seconds will result in a non-smooth and choppy video, while a playback time which is longer than 2.0 seconds will yield a lengthy and slow-paced one. Therefore, when given a clustered shot set $\boldsymbol{W}$, the video sequence with the minimum redundancy measure is the one in which all the shot clusters have a uniform occurrence probability and an equal time length of 1.5 seconds.

Based on these criteria, our video summarization method creates video summaries using the following steps:

1) Segment the video into individual camera shots using the method in [50]. The output of this step is a shot set $\boldsymbol{S}=\{s_1, s_2, \ldots s_i, \ldots, s_n\}$, where s_i represents the i^{th} shot detected and n is the total number of shots detected.
2) Group the camera shots into a clustered shot set $\boldsymbol{W}$ based on their visual similarities.

The similarity between two detected shot can be represented by their key frames. For each shot $s_i \in \boldsymbol{S}$, we choose a key frame f_i as the representative frame of that shot. We choose the middle frame of a shot as the key frame, other than at the two ends of a shot, because the shot boundaries commonly contain transition frames. When comparing the visual similarities of two different shots, we calculate the difference between two key frames related to these two shots using color histograms:

$$D_v(i,j) = \left| \sum_{e=Y,U,V}^{k=1..n} h_i^e(k) - h_j^e(k) \right| \tag{3-14}$$

where h^e_i , h^e_j are the histograms of the key frame *i* and *j,* respectively.

The main difficulty here is that the optimal number of the clusters needs to be determined automatically. To solve this problem, we use the adaptive shot clustering method described in [50]. After this step, the original video sequence can be described by the clustered shot set $\boldsymbol{W} = \{w_1, w_2, \ldots, w_k\}$.

3) For each cluster, find the shot with the longest length, and use it as the representative shot for the cluster.
4) Discard the clusters whose representative shots are shorter than 1.5 seconds. For those clusters whose representative shots are longer than 1.5 seconds, we curtail those shot to 1.5 seconds by truncating the first 1.5 second visual content from those shots.
5) Sort the representative shots of all the clusters by the time code.

Now, we have the representative shot set $\boldsymbol{U}=\{u_1, u_2, \ldots, u_m\}$, where $m \leq n$, and n is the total number of the shots in set $\boldsymbol{S}$.

3.3.3 Music/Video Alignment

The final task to create a music video summary is to align the image segments in the video summary with the associated music segments in the music summary. According to [62], the visual and audio content combination in the music video can be divided into two categories: the polyphonic structure and homophonic structure. Based on the analysis in Section 3.3.1, we currently use the same alignment scheme for these two music video structures. As mentioned in Section 3.3.1, the goal of alignment is to make the summary smooth and natural, and generate a summary which maximizes the coverage for both music and visual content of the original music video without sacrificing music or visual part.

Assume that the whole time span L_{sum} of the video summary is divided by the alignment into P partitions (required clusters), and the time length of partition i is T_i. Because each image segment in the video summary must be at least L_{min} second long (a time slot equals to one L_{min} duration), partition i will provide N_i time slots, as shown in Figure 3-8:

$$N_i = \lceil T_i / L_{\min} \rceil \tag{3-15}$$

and hence the total number of the available time slot becomes :

$$N_{total} = \sum_1^P N_i \tag{3-16}$$

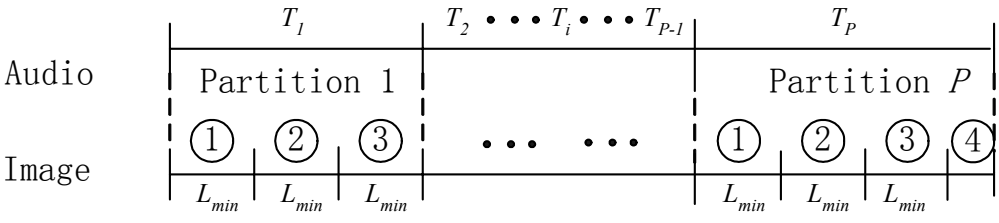


Figure 3-8: Alignment operations on image and music

Recall that for each partition, the time length of the music sub-summary lasts for 3~5 seconds, and the time length of a shot is 1.5 seconds. The situation that the sum of the visual shots exceeds the sub-summary of the music in the partition will appear. We handle this situation by constraining the last shot of that partition to fit the sub-summaries of the music. As shown in Figure 3-8, T_P is the time length of partition P and lasts for 5 seconds. Four shots are found to fill in this partition, each of which lasts for 1.5 seconds. The total length of the video sub-summary is 6 seconds, which is longer than the music sub-summary. Thus, we curtail the last shot ④ to fit the video sub-summary to the music sub-summary.

Therefore, the alignment problem can be formally described as:

Given:

1. An ordered set of representative shots $\boldsymbol{U}=\{u_1, u_2,\ldots, u_m\}$, where $m \le n$, and n is the total number of shots in the shot set $\boldsymbol{S}$.

2. P partitions and N_{total} time slots.

To extract:

P sets of output shots subset , $\boldsymbol{R}=\{R_1, R_2,\ldots,R_P\}$ which best match between the shot set $\boldsymbol{U}$ and N_{total} time slots.

where:

P=Number of partitions

$R_i=\{r_{i1},\ldots,r_{ij},\ldots,r_{iNi}\} \subseteq \boldsymbol{U}$, $i=1, 2, \ldots, P$; $N_i=\lceil T_i / L_{\min} \rceil$.

Shots $r_{i1},\ldots,r_{ij},\ldots,r_{iNi}$ are optimal shots selected from the shot set $\boldsymbol{U}$ for the i-th partition.

The constraints:

1) At the first time slot of each partition, the root shot filled in that time slot will be the shot corresponding to that time slot in time domain

2) Within each output shots subset R_i, shots should have the highest similarity in terms of temporal and visual features to ensure the summary smooth and natural.

By a proper reformulation, this problem can be converted into a Minimum Spanning Tree (MST) problem [64]. Let $\boldsymbol{G} = (\boldsymbol{V},\boldsymbol{E})$ represent an undirected graph with a weighted edge set $\boldsymbol{V}$ and a finite set of vertices $\boldsymbol{E}$. The MST of a graph defines the lowest-weight subset of edges that spans the graph in one connected component. To apply the MST to our alignment problem, we use each vertex to represent a representative shot u_i, and an edge $\boldsymbol{e}_{ij}=(u_i, u_j)$ to represent the similarity between the shot u_i and u_j. The similarity here is defined as the combination of time similarity and visual similarity. The similarity function is defined as follows:

$$e_{ij} = (1-\alpha)T(i,j) + \alpha D(i,j) \tag{3-17}$$

where α is a weight coefficient, which is set in advance according to the priority given to the visual similarity and the time similarity. The lower α is, the lower priority for visual similarity and the higher priority for time similarity, and vice verse. In our experiment, since the time similarity which indicates time synchronization information is much more important than the visual similarity, we give the time similarity a higher priority. We set α =0.2 for all testing samples.

$D(i,j)$ and $T(i,j)$ in Equation (3-17) represent the normalized visual similarity and time similarity, respectively.

$D(i,j)$ is defined as follows:

$$D(i,j) = D_v(i,j)/\max(D_v(i,j)) \tag{3-18}$$

where $D_v(i,j)$ is the visual similarity calculated from Equation (3-14). After normalized, $D(i,j)$ has a value range from 0 to 1.

$T(i,j)$ is defined as follows:

$$T(i,j) = \begin{cases} 1/(F_j - L_i) & L_i < F_j \\ 0 & otherwise \end{cases} \tag{3-19}$$

where L_i is the index of the last frame in the i^{th} shot, and F_j is the index of the first frame in the j^{th} shot. Using this equation, the closer two shots are in the time domain, the higher time similarity value they have. Value $T(i,j)$ varies from 0 to 1, and the biggest value of $T(i,j)$ achieves when shot j just follows shot i and there is no other frames between these two shots. Thus, we can create similarity matrix $\boldsymbol{\Phi}$ for all shots in the representative shots set $\boldsymbol{U}$, and the i,jth element of $\boldsymbol{\Phi}$ is e_{ij}.

For every partition R_i, we generate a MST based on the similarity matrix $\boldsymbol{\Phi}$.

To create content rich audio-visual summary, we propose the following alignment operations:

1) Summarize the music track of the music video using the method described in Section 3.2. The music summary consists of several partitions, each of which lasts for 3 to 5 seconds. The total duration of the summary is about 30 seconds. We can get the music summary by adjusting the parameters of the algorithm described in the previous section.

2) Divide each music partition into several time slots, each of which lasts for 1.5 seconds.

3) For each music partition, we find the corresponding image segment as follows: In the first time slot of the partition, find the corresponding image segment in the time domain. If it exists in the representative shot set $\boldsymbol{U}$, assign it to the first slot and delete it from the shot set $\boldsymbol{U}$; if not, identify it in the shot set $\boldsymbol{S}$, and find the most similar shot in shot set $\boldsymbol{U}$ using similarity measure defined in Equation (3-14). We then take this shot as the root, apply the MST algorithm to it, find other shots in the shot set $\boldsymbol{U}$, and fill them in the subsequent time slots in this partition.

Figure 3-9 illustrates the alignment process, where A(t_i ,τ_i) and I(t_i ,τ_i) denote audio and visual segments that start at time instant t_i and last for τ_i seconds, respectively. The length of the original video program is 40 seconds. Assume that the music summarization has selected three partitions A(0,3), A(13,5) and A(23,4), and the shot clustering process has generated twelve shot clusters shown in Figure 3-9. As the music summary is generated by A(0,3), A(13,5) and A(23,4), we divide this twelve-second summary into nine time slots. For each slot, we assign a corresponding shot. For the first partition, we assign shot ① to time slot *a* and shot ② to time slot *b*, respectively. When we assign a shot to the time slot c, there is no corresponding image segment in the time domain. According to our alignment algorithm, we choose shot ④ which is a most similar shot in line with the time index in the shot set $\boldsymbol{S}$. Then, based on shot ④, we apply the MST algorithm to find other shots for the second

partition. For the third partition, in the first time slot *g*, because the corresponding visual segment ⑦ has been used by other slots, we have to find a most similar shot to shot ⑦ in the shot cluster set ***U***. Based on the algorithm described above, we find shot ⑧. We then apply the MST algorithm to find other two shots from this partition.

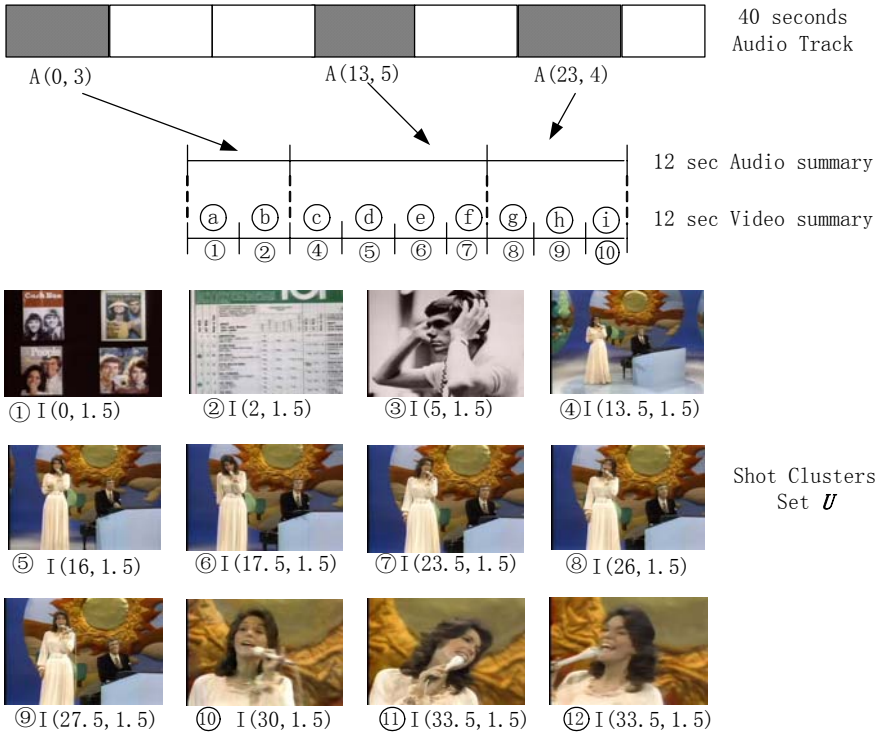


Figure 3-9: An example of the audio-visual alignment

In this way, our proposed summarization scheme can maximize the coverage for both music and visual content of the original music video without sacrificing either of them. In the created summary, the visual content may not be strictly synchronized with the music. As we mentioned before, the experiment on human perception shows that the visual attention is not sensitive to the audio-visual asynchrony [63]. Therefore, within the range of human perception acceptance, the minor deviation for the visual content from the music is allowed. In our method, by giving the time similarity between the shots a high priority (adjust weight α), we can control the visual deviation from the music in the range of human perception acceptance.

Some experimental results for our proposed music summarization method can be found in section 5.2, and the music video summarization examples can be viewed at http://www.comp.nus.edu.sg/~shaoxi/Vsum/vsum1.htm

3.4 Summary

In this chapter, a novel music summarization scheme was presented. In addition, as an extension of the music summarization, the music video summarization scheme was also proposed based on the music summarization. The main contribution includes:

- An adaptive clustering algorithm was proposed to adjust the overlapping rate of the music signal segmentation window, which aims to optimally group the music frames to get the good summarization results.
- Considering the different roles pure instrumental music and vocal music play in the song, we propose a machine learning approach to identify vocal music vs. pure music segments and use this to select audio segments for summaries.
- Based on summary for music track, we proposed an algorithm to align the structured visual shots with the generated music track summary to generate a video summary for music video. The proposed alignment algorithm maximizes the coverage of important audio segments along with important video segments.

Music summarization aims to structure the individual music piece in database according to its intrinsic repeating patterns, and therefore is very helpful in music database management applications such as genre classification and retrieval. On

one hand, music structure information obtained in the music summarization can be utilized in music genre classification. For example, some music genres have a fairly rigid format, others are more flexible. Therefore, using the music structure information, we can roughly classify music genre at a coarse-level. On the other hand, the representative segments obtained by music summarization contain most memorable information for human beings. In the retrieval process, giving the high priority to these segments will significantly reduce the search space. As a result, interaction with large music databases can be made simpler and more efficient.

Real World Music Retrieval by Humming 4

After the music database has been structured, it should be easily accessed by users. Music Information Retrieval (MIR) is primarily concerned with efficient content-based searching and retrieval of music information from music database. Currently, the music information retrieval systems can be divided into different categories as shown in the Table 4-1 [65], based on the different representations of the query side and the database side. As the table shows, in the category marked "Solvable", both the query and database side are symbolic format, such as MIDI. Then the music retrieval problem can be converted into text-based retrieval problem, which can be solved by methods derived from text searching techniques. As for retrieving from monophonic[1] acoustic database with monophonic or symbolic queries, such music retrieval systems are of little practical value, since there is not much monophonic audio data available.

[1] The definition of monophonic music and polyphonic music can be found in [4].

Table 4-1.Classification of music information retrieval system

Database \ Query		Symbolic	Acoustic	
			Monophonic	Polyphonic
Symbolic		Solved	QBH1	?
Acoustic	Monophonic	Solvable, but may not be interested in practice		?
	Polyphonic	?	QBH2	QBE

We select query by humming as the query approach since humming is the most natural way to formulate music queries for people who are not trained or educated with music theory [66]. However, the natural music format for most of the music database is not symbolic music, but raw polyphonic audio. Therefore, from the usability point of view, the investigation for query by humming based on polyphonic raw music database is becoming important and necessary. In order to distinguish from the QBH for Symbolic database system (denoted as QBH1), we denote QBH for polyphonic database system as QBH2, to which the issue that we try to solve in this thesis belongs.

In the category mark with QBE in the Table 4-1, namely Query By Example, such MIR system retrieve from a polyphonic audio database in response to a similar audio query. The most relevant dissertation to this work is [65]. The category marks with question marks remain open problems. Due to the lack of general-purpose polyphonic transcription algorithm, we cannot expect to solve these problems by reducing them to monophonic or symbolic cases.

In this chapter, we first provide an overview of current state for query by

humming based on MIDI database and the state of the art for problem of separating mixtures in the real world environment, which can be modeled as convolutive mixtures. Considering that the vocal content separation is important for our specific problem, we put emphasis on it in this chapter: In section 4.2, we provide background theory for the independent source separation for separating mixtures in the real world environment, and then in section 4.3 ,based on the fact that the vocal singing voice and background music are two heterogeneous signals, we propose a statistical learning based method to solve the permutation inconsistency problem in Frequency Domain Independent Component Analysis (FD-ICA), which is an unsolved problem in convolutive mixture separation. Based on the separated vocal content, some refinements to convert it to the standard music representation (sequence of notes) can be found in section 4.4. Finally, we summarize this chapter with section 4.5.

4.1 Related Work

Several methods to solve QBH1 have been proposed in the past years. Ghias [67] reported surprisingly effective retrieval using query melodies that have been quantized to three levels, depending on whether each note was higher, lower, or similar as the previous one. Besides simplifying the pitch extraction, this allowed for less-than-expert singing ability on the part of the user. MaNab [68] used flexible string-matching algorithm to locate similar melodies located anywhere in a piece and provides detailed design information with a prototype system encompassing all the aspects of a music retrieval facility. The system transcribed acoustic input, typically

sung or hummed by the user, and retrieved music, ranked by how closely it matched the input. In order to take into account human inaccuracies of recall and of performance, the errors that people make in remembering melodies and in singing were modeled. The flexible retrieval mechanisms that were tailored to the errors actually encountered in practice were devised. However, it was not applicable to a very large database, since searching the whole database for each query will lead to more and more expensive computations with the database growing. In [69], music melody was represented by 4 types of segments according to the shape of the melody contour, the associated segment duration and segment pitch. Song retrieval was conducted by matching segments of melody contour. The basic idea for above methods is similar. Pitch contour of the hummed query is detected and pitch changes are converted into strings according to the direction and/or magnitude of the pitch change. Similarly, the melody contour of MIDI is also converted into strings. String matching algorithms are employed to do similarity retrieval. The string matching approach requires precise detection of individual notes (onset and offset) out of the hummed query. However, it is not uncommon for people to substitute a long note with several short notes with same pitch value while humming a tune. When there are tied notes in the melody, it is likely that incomplete notes will be detected. The string matching result would suffer drastically when the error in note detection is not minor. To deal with above-mentioned issue, Zhu [70] proposed a new slope-based query-by-humming approach, in which the retrieval is robust to the inaccuracy in query and the use of the metronome is eliminated. A pitch tracking method was used

to construct the melody curve from a user's humming. Curve features like melody slope pitch range, time duration, and note changes in the slopes were extracted from melody curves. A melody slope matching algorithm was applied to conduct retrieval. When rhythm and pitch interval is considered, more complex similarity measure and matching algorithm should be used. In [71] the author employed two-dimensional augmented suffix tree to search the desired song, and in [72], a new distance metrics between query and songs is proposed.

All the approaches for the music retrieval mentioned above are based on MIDI files database, which are easy to be represented by a symbolic sequence. To make content-based music retrieval more applicable, the retrieval task should be extended to the music files of real world digital audio recordings.

One direction to solve this problem is to detect the vocal melody contour directly from the polyphonic music. Most current pitch detection methods for polyphonic music are limited to pitch detection in modest noise [73] [74] [75]. Recently, some algorithms for predominant fundamental frequency tracking have been investigated. Goto [76] employs a Maximum A Posterior Probability (MAP) estimation to estimate the relative dominance of every fundamental frequency and the shape of harmonic structure tone models, but the performance on tracking predominant vocal pitch mixed with significant broadband noise interference is not clear. In [77], to avoid the problem of extracting exact pitch information from polyphonic raw audio in signal level, the author instead proposed to represent the melody information in a statistic way, which of course cannot guarantee the retrieval accuracy.

The other intuitive solution to this extension is to develop some algorithms to extract a certain representation from the polyphonic music in the database, and then all the current technology of query by humming based on monophonic representation can be used. If we can, for example, separate the vocal content from the background music, then the QBH2 problem can be converted to QBH1 problem. However, the design of such an algorithm is difficult, since the background music interferes with vocal content both in time domain and frequency domain. Traditional Independent Component Analysis (ICA) [78] algorithm is not applicable as it assumes the independent sources are mixed instantaneously, while common polyphonic music has two channels (mixtures) which generally are convolutions of the two sources (singing voice and background music). In [79] [80], time domain algorithms for convolved mixture separation were proposed using the maximum entropy cost function. These time domain algorithms work well for small length mixing filters, but when it comes to real time implementations with realistically long filters, they will be unrealizable because of lack of computational efficiency. In addition, updating one coefficient for a particular filter will account for the already updated preceding filter coefficients, which prevents the convergence to the optimal filter coefficients. Therefore, it is intuitive to move from the time domain solution to the frequency domain as the convolution in the time domain is multiplication in the frequency domain and apply ICA methods for instantaneous mixtures in each frequency bin. In this way, the unmixing matrix in each frequency bin is independent and will not interfere with each other. However, since we obtain the unmixing matrix in each frequency bin

independently, arbitrary permutation of unmixing matrix in certain frequency bin will lead to the same value for maximum entropy cost function. This seems to be a serious problem as only consistent permutations for every frequency bins will correctly reconstruct the sources. This problem is called *permutation inconsistency* problem [81] in FD-ICA. Some channel-based frequency coupling methods [82] were proposed to solve this problem by placing smoothness constraints across the frequency bin. However, such constraints reduce the available degrees of freedom to reconstruct the sources. On the other hand, alternative approaches called sources based frequency coupling were proposed in [83] [84]. They tried to solve the problem by exploiting the relationship between the reconstructed sources at a frequency bin and the original sources in the time domain. However, the basic assumption that one source is louder at certain time slot may be valid for convolutive mixtures of speech signals, but may not always valid for the mixtures of singing voice and background music.

In this thesis, we present an approach to practical QBH2 music retrieval system for two channel polyphonic music.

4.2 Background Theory for Blind Source Separation

We all know the problem: we are in a party, people are talking and it is quite hard to understand each other because of all the interference. You can imagine, it is even harder for a machine to separate individual speeches. This is the well known problem often referred to as the “Cocktail Party Problem” [85].

The solution to these kinds of problems is called “Blind Source Separation” (BSS)

[86]. Blind source separation attempts, as the name states, to separate a mixture of signals into their different sources. The word "blind" is used because we have no prior knowledge about the statistics of the source in general.

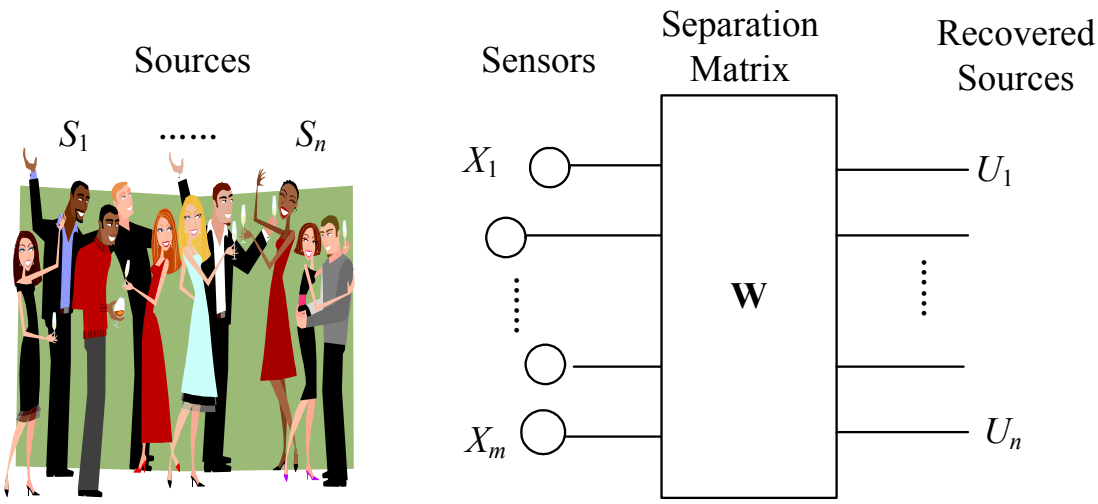


Figure 4-1: The illustration of Cocktail Party Problem and BSS

Figure 4-1 illustrates the cocktail party problem and the Blind Sources Separation to it. As the figure shows, we assume that we have n sources, $S_{i,}$, which transmit signals that after propagation in an arbitrary medium are measured by m sensors , and the signals that are measured by these sensors will be called X_i. The mapping from S_i to X_i is an unknown linear function f_i so that:

$$X_i = f_i(S_1, S_2, \cdots, S_n) \tag{4-1}$$

Using linear algebra notation, we can rewrite the above equation at the sensor side in a more elegant form as:

$$\mathbf{X}(t) = \mathbf{A} \cdot \mathbf{S}(t) \tag{4-2}$$

Where $\mathbf{X}(t)=[X_1(t),\cdots,X_m(t)]^T$, $\mathbf{S}(t)=[S_1(t),\cdots,S_n(t)]^T$, and $\mathbf{A} \in \Re^{m\times n}$ is an unknown invertible matrix which we will can the mixing matrix. The task of Blind Sources Separation is to find a separation matrix **W**, which is expected to be the inverse (or

pseudo-inverse) of the mixing matrix **A**, to recover the original sources. The unmixing equation can be defined as:

$$\mathbf{U}(t) = \mathbf{W} \cdot \mathbf{X}(t) \tag{4-3}$$

The output $\mathbf{U}(t)=[U_1(t), \cdots, U_n(t)]^T$. Once $\mathbf{W} \approx \mathbf{A}^{-1}$, and then we would have $\mathbf{U}(t) \approx \mathbf{S}(t)$.The notation used here will be adopted for the remaining of the chapter.

4.2.1 Different Approaches for BSS

- Independent Component Analysis method

Blind source separation by Independent Component Analysis (ICA) has received attention because of its potential applications in signal processing. It is an information theoretic approach, and used to find a linear non-orthogonal co-ordinate system in any multivariate data. The directions of the axes of this coordinate system are determined by both the second and higher order statistics of the original data. The goal is to perform a linear transformation which makes the resulting variables as statistically independent from each other as possible. A good introduction to ICA can be found in [87] [88].

- Bayesian Approach

It provides a probabilistic approach to estimation and inference. It is based on the assumption that the quantities of interest are governed by probability distributions, and that optimal decisions can be made by reasoning about these probabilities together with the data. Bayesian approaches[89][90] provide a framework for learning algorithm that manipulate probabilities directly as well as for learning algorithms that

do not explicitly manipulate probabilities. For example, in [89], the author provides a Bayesian approach to source separation. The basic idea of this method: forming a model that describes a particular source separation problem that can be described by a simple linear model consisting of a mixing matrix **A** and the source signal time sequence **S**(*t*). The observation data can be described as **X**(*t*).

$$P(model \mid data, I) = \frac{P(data \mid model, I)P(model, I)}{P(data, I)} \tag{4-4}$$

With data = **X**(*t*); model = **A**, **S**(*t*); I = any priori knowledge.

The maximum likelihood estimation algorithm can be used to find the parameters that maximize the *P* (*model* | *data*, *I*) in the above equation.

This approach, however, provides a framework for learning algorithm, other than the learning algorithm itself. Therefore, it is more suitable to model selection problem than providing solution to the model [91].

- TDSS(Temporal Decorrelation Source Separation)

It uses the temporal structure of signals in order to compute the time-delayed 2nd order correlation for the source separation [92][93][94]. The best results are achieved if the autocorrelations are as different as possible. The main point of TDSS is to diagonalize the covariance matrix $\mathbf{C}_0 =< \mathbf{X}(t) \cdot \mathbf{X}(t-\tau)^T >$ for $\tau = 0$ (no delay) and at the same time diagonalize the covariance matrix for a given delay $\mathbf{C}_\tau =< \mathbf{X}(t) \cdot \mathbf{X}(t-\tau)^T >$.This leads to an eigenvalue problem as described in[92]:

$$(\mathbf{C}_0 \, \mathbf{C}_\tau^{-1})\mathbf{A} = \mathbf{A}(\mathbf{\Lambda}_0 \mathbf{\Lambda}_\tau^{-1}) \tag{4-5}$$

where $\mathbf{\Lambda}$ is the diagonal matrix with elements that are the eigenvalues of the

corresponding covariance matrix. The TDSS algorithm can be extended to a matrix of filters [95]. The advantage of TDSS over the traditional Independent Component Analysis is that it computationally fairly inexpensive. However, the disadvantage is that this approach assumes the minimum-phase mixing filters, which limit it usefulness to the mixtures in the real world environment [88].

- Blind Separation of Disjoint Orthogonal signal

It uses only 2 mixtures of N sources, but the sources have to be pair-wise disjointly orthogonal in time frequency representation [96] [97] [98] [99]. The algorithms are based on the Short Time Fourier Transform. The major problem for this approach is that it is based a rather strong assumption, which considers that the time-frequency representation of different sources do not overlap. This assumption may hold for the speech mixtures, but is not true the music and vocal singing mixture.

- Principle Component Analysis (PCA)

It uses second order methods in order to reconstruct the signal in the mean square error sense. The results are independent of the second order statistics. In some areas, this is called KL-transform [100]. The basic idea in PCA is to find the components $S_1(t),\cdots, S_n(t)$ so that they explain the maximum amount of variance possible by n linearly transformed component. As mentioned in [87], the purpose of PCA is to find a *faithful* representation of the data. This is in contrast to most high order methods such as ICA which try to find a *meaningful* representation. A good comparison for ICA and PCA can be found in [88].

The ICA method is adopted in this thesis since it approximates the way how the

human brain solves the problem. It has been hypothesized that the brain is a ultimately a sophisticated statistical engine, where thought is modeled by statistical inference rather than logic and learning results from accumulation of massive date from interactions with the world. So a statistical method that claims to model information decomposition and encoding in the brain is certainly worthy of examination.

4.2.2 Traditional ICA to Solve Instantaneous Mixtures

Independent Component Analysis (ICA) is a statistical method to separate complex datasets into independent sub-parts. ICA exploits the non-gaussianity of source signals and assumes statistical independence of the separated signals to perform separation. Similar ICA learning rules for separating instantaneous mixtures have been developed from a number of different view points: Informax [78] , Minimizing Kullback-Leibler (KL) divergence [101]. In this section, we will describe these approaches briefly, since separating instantaneous mixtures is tightly related to our specific problem, and acts as the basis of real world mixtures separation.

- Informax approach:

In [78], Bell etc. proposed a simple learning algorithm for a feed forward neural network that blindly separates linear instantaneous mixture **X** of independent sources using information maximization. The structure they proposed in 2 by 2 case has been show in Figure 4-2. They show that maximizing the joint entropy $H(\mathbf{Y})$ of the output of the neural processor can approximately minimize the mutual information among the

output components $Y_i = g_i(U_i)$, where $g_i(U_i)$ is an invertible monotonic nonlinearity function (so-called activation function), and **U=WX**.

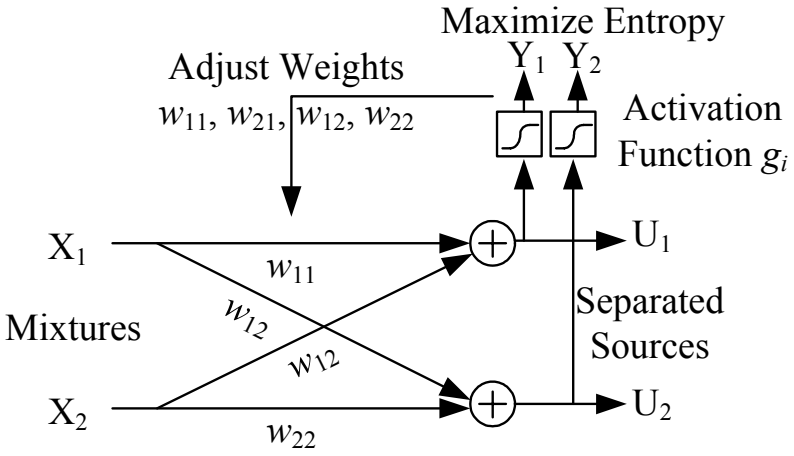


Figure 4-2: Separation network for instantaneous mixtures

The joint entropy (Appendix C.2) at the output of neural network is:

$$H(Y_1, Y_2) = H(Y_1) + H(Y_2) - I(Y_1, Y_2) \tag{4-6}$$

where $H(Y_i)$ are the marginal entropies (Appendix C.1) of the outputs and $I(Y_1, Y_2)$ is their mutual information (Appendix C.5). Maximizing the $H(Y_1, Y_2)$ consists of maximizing the marginal entropies and minimizing the mutual information. The output $\mathbf{Y} = (Y_1, Y_2)^T$ are amplitude-bounded random variables and therefore the marginal entropies are maximum for a uniform distribution of Y_i. As author pointed out in the paper, maximizing the joint entropy will also minimizing the mutual information $I(Y_1, Y_2)$, on the condition that Probability Density Function (PDF) of the independent sources are super-gaussian, to which the most real world acoustic signals belong. When $I(Y_1, Y_2)$=0, the joint entropy is the sum of marginal entropies:

$$H(Y_1, Y_2) = H(Y_1) + H(Y_2) \tag{4-7}$$

Therefore, the maximal value for $H(Y_1, Y_2)$ is achieved when the mutual information among the bounded random variable Y_1, Y_2 is zero and their marginal distribution is

uniform. There are two sets of parameters that determine the maximum joint entropy: The nonlinearity $Y_i = g_i(U_i)$ and the synaptic efficacies **W.** Bell et. al. choose the nonlinearity to be a fixed logistic function, which is equivalent to assuming a prior distribution of the sources: A super-gaussian distribution[2] with heavy tails and a peak centered at the mean. The only remaining parameters to adopt are the synaptic weights and they can be obtained by the following learning rule (Full derivation can be found in Appendix D.1):

$$\Delta \mathbf{W} \propto [\mathbf{W}^T]^{-1} - 2 \cdot \tanh(\mathbf{U}) \cdot \mathbf{X}^T \tag{4-8}$$

Where tanh(·) is the hyperbolic tangent function. Bell used this algorithm successfully for separating instantaneous mixtures of up to 10 sources.

- Minimizing Kullback-Leibler (K-L) divergence Approach

Amari et. al.[101] used the Kullback-Leibler (K-L) (measure of PDF similarity for different statistical variables, see detail information in Appendix C.4) as the start point, and derived the similar learning rule as Bell's. The basic idea behind this approach is to choose the mutual information $I(U_i, U_j)$ between the random variable U_i, U_j constituting any two components of the output vector **U**. When in the ideal case, $I(U_i, U_j)$ is zero, the component U_i, U_j are statistically independent. This would therefore suggest minimizing the mutual information between every pair the random variables constituting the output vector **U**. This objective is equivalent to minimizing the K-L divergence between following two distributions:

[2] For a random variable *X*, it's fourth-order cumulant, called Kurtosis, which can be expressed as

$$\text{Kurt}(x)=\text{E}\{x^4\}-3(\text{E}\{x^2\})^2$$

Kurtosis can be considered as a measure of the non-gaussianity of *X*. For a gaussian variable, Kurtosis is zero. Distributions of positive (negative) Kurtosis are called super-gaussian (sub-gaussian). Super-gaussian distribution typically has heavy tails and a peak centered at the mean while sub-gaussian distribution has flatter density with lighter tails.

1) The PDF $f_{\mathbf{U}}(\mathbf{U},\mathbf{W})$ parameterized by $\mathbf{W}$

2) The corresponding factorial distribution $\overline{f}_{\mathbf{U}}(\mathbf{U},\mathbf{W})$ which is defined as:

$$\overline{f}_{\mathbf{U}}(\mathbf{U},\mathbf{W}) = \prod_{i=1}^{n} \overline{f}_{U_i}(U_i,\mathbf{W}) \tag{4-9}$$

Where $\overline{f}_{U_i}(U_i,\mathbf{W})$ is the marginal PDF of random variable U_i.

Using the K-L divergence as the cost function, synaptic weights can be obtained by the following learning rule (see Appendix D.2)

$$\Delta\mathbf{W} \propto [\mathbf{W}^T]^{-1} - \varphi(\mathbf{U})\cdot\mathbf{X}^T \tag{4-10}$$

where φ(x) is the activation function and $\varphi(x) = \frac{3}{4}x^{11} + \frac{25}{4}x^9 - \frac{14}{3}x^7 - \frac{47}{4}x^5 + \frac{29}{4}x^3$.

The activation function is not unique and other activation function such as tanh can also be used depending on the distributions of the sources. It should be noted that by replacing Amari's activation function with the hyperbolic tangent we get more stable learning since φ(x) is not bounded and large learning rates result in numerical overflows.

As proposed by Amari et. al., a much more efficient way to maximize the joint entropy is to follow the 'natural' gradient. The natural gradient rescales the normal gradient space by right multiplying $\mathbf{W}^T\mathbf{W}$ in the both sides of equation 4-11, which gives the following:

$$\Delta\mathbf{W} \propto [\mathbf{I} - \varphi(\mathbf{U})\cdot\mathbf{U}^T]\cdot\mathbf{W} \tag{4-11}$$

By performing the descent using natural gradient, convergence is significantly faster and more stable. In addition to good convergence behavior, there is also increased efficiency since the learning rule does not include a matrix inversion operation.

Unfortunately, the instantaneous mixture is rather incomplete in the real world

situation, due to the extensive filtering imposed between sources and the sensors. Instead, we consider the convolutive mixture separation problem.

4.2.3 Convolutive Mixture Separation Problem

According to [102], the music companies produce their music products in basically two stages. First, sound from each individual instrument is recorded in an acoustically inert studio on a single track of a multi-track tape recorder. Then, the signals from each track are manipulated by the sound engineer to add special audio effects and are combined in a mix-down system to finally generate the stereo recording on a two-track recorder. The audio effects are artificially generated using digital signal processing techniques and these digital signal processing techniques can be considered as direct filter and cross filter placed between the sources and output channels. Therefore, the generation of song clips can be modeled as the Figure 4-3(a) and the mixture process can be modeled by following equation:

$$X_1(n) = A_{11}(n) * S_1(n) + A_{21}(n) * S_2(n) \quad \text{(4-12-a)}$$

$$X_2(n) = A_{12}(n) * S_1(n) + A_{22}(n) * S_2(n) \quad \text{(4-12-b)}$$

where $\boldsymbol{x}_1$(n) and $\boldsymbol{x}_2$(n) represent two channels of the polyphonic music respectively, $\boldsymbol{s}_1$(n) and $\boldsymbol{s}_2$(n) represent two sources respectively. A_{11} and A_{22} denote the P points direct filter between sources and channels, and A_{12}, A_{21} denote the P points cross filter between sources and channels, respectively. Then the basic problem can be described as follows:

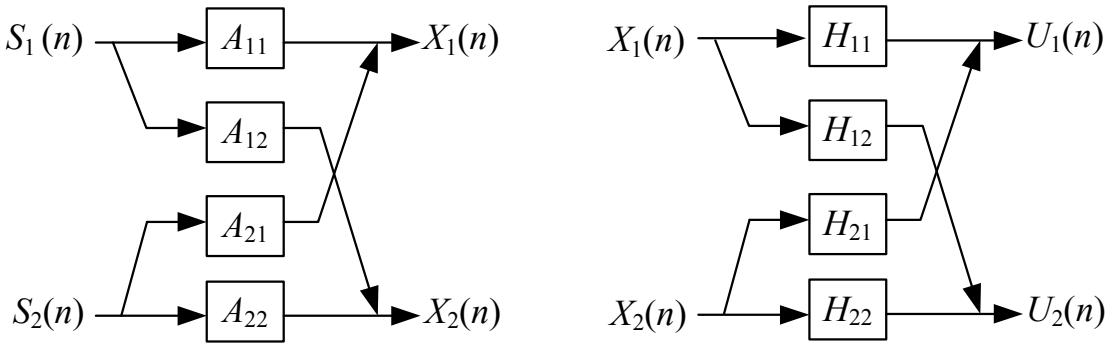
Given the observed channels $X_1(n)$ and $X_2(n)$, we expect to find a filter matrix **H**

to separate the independent sources $S_1(n)$ and $S_2(n)$ from the observed mixtures $X_1(n)$ and X_2(n). The unmixing process can be modeled by following equation:

$$U_1(n) = H_{11}(n) * X_1(n) + H_{21}(n) * X_2(n) \qquad \text{(4-13-a)}$$

$$U_2(n) = H_{12}(n) * X_1(n) + H_{22}(n) * X_2(n) \qquad \text{(4-13-b)}$$

Our goal is to obtain the separated sources $U_1(n)$ and $U_2(n)$ to approximate the original sources $S_1(n)$ and $S_2(n)$ as closely as possible. The unmixing process can be illustrated in Figure 4-3(b).



(a) Mixing process (b) unmixing process

Figure 4-3: The convolutive source separation problem

In [79][103], time domain algorithms for convolved mixture separation were proposed using the maximum entropy cost function. These time domain algorithms work well for small length mixing filters, but when it comes to real time implementations with realistically long filters, they will be unrealizable because of lack of computational efficiency. In addition, updating one coefficient for a particular filter will account for the already updated preceding filter coefficients, which prevents the convergence to the optimal filter coefficients[81]. In [104], the author employ another time domain architecture so called feed back architecture, to perform the convolved mixture separation, but the computational inefficiency problem is still

unsolved. Therefore, it is intuitive to move from time domain solution to the frequency domain, since the time-domain convolutive mixture can be transformed to an instantaneous mixture in the frequency domain by computing its Q-points Short-Time Fourier Transform (STFT):

$$\mathbf{X}(f,t_s) = \mathbf{A}(f)\cdot\mathbf{S}(f,t_s) \quad f = 1,2,\ldots,Q \tag{4-14}$$

where t_s is the block index, $\mathbf{X}(f,t_s)=(\mathbf{X}_1(f,t_s)\ ,\ \mathbf{X}_2(f,t_s))^T$ represents the Short-Time Fourier Transform of two observed channels and $\mathbf{S}(f,t_s)=(\mathbf{S}_1(f,t_s)\ ,\ \mathbf{S}_2(f,t_s))^T$ denotes the STFT of two independent sources. $\mathbf{A}(f)$ denotes a 2×2 instantaneous complex matrix at the frequency f. Then the problem can be defined as the estimation of an unmixing matrix $\mathbf{H}_f \approx \mathbf{A}^{-1}(f)$ for each frequency bin. This unmixing matrix $\mathbf{H}_f$ can be obtained by extending the real value blind source separation approach for instantaneous mixture [78] to the complex domain. The estimation process can be considered as to obtain a Maximum Likelihood solution separately for each frequency bin by maximizing the following criteria function [78]:

$$\log p(\underline{\mathbf{X}}(f,t)\big|\mathbf{H}_f) = E\{\log\ p(\mathbf{U}(f,t))\} + \log(\det(\mathbf{H}_f)) \tag{4-15}$$

where $\mathbf{U}(f,t)$ represents the separated sources, and $E(\cdot)$ represents the expectation. According to [78][81], to optimize the criteria function, the learning function for unmixing matrix $\mathbf{H}_f$ derived from Eq.(4-15) can be expressed as:

$$\Delta\mathbf{H}_f \propto (\mathbf{I} - \varphi(\underline{\mathbf{U}}(f,t))\cdot\underline{\mathbf{U}}(f,t)^{Herm})\mathbf{H}_f \tag{4-16}$$

where $(\cdot)^{Herm}$ denotes the Hermitian transposition and $\varphi(\cdot)$ is the activation function proposed in [81]:

$$\varphi(z) = \tanh(z_R) + i \cdot \tanh(z_I) \tag{4-17}$$

where z_R is the real part of z and z_I is its imaginary part..

Figure 4-4 illustrates the blind source separation process in the frequency domain.

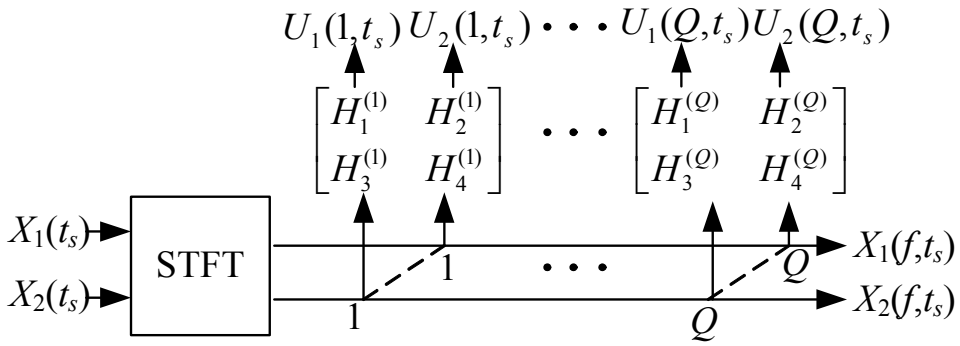


Figure 4-4: Frequency domain blind source separation

We obtained the unmixing matrix independently in each frequency bin and the arbitrary permutation of any particular unmixing matrix in certain frequency bin will not change the value of the criteria function. As a result, the algorithm produces different permutations of separated sources along the frequency axis and the sources still remain mixed. This problem is called *permutation inconsistency* problem [81] in FD-ICA. Some channel-based frequency coupling methods [81] [82] [105] were proposed to solve this problem by placing smoothness constraints across the frequency bin. However, such constraints reduce the available degrees of freedom to reconstruct the sources. On the other hand, alternative approaches called sources based frequency coupling were proposed in [83] [84] [106] [107]. They tried to solve the problem by exploiting the relationship between the reconstructed sources at a frequency bin and the original sources in the time domain. However, the basic assumption that one source is louder at certain time slot may be valid for convolutive mixtures of speech signals, but may not always valid for the mixtures of singing voice

and background music.

4.3 Our Proposed Permutation Inconsistency Solution

To solve the permutation inconsistency problem, we propose to use a statistical learning based approach to classify the output sources in each frequency bin and keep the output sources consistent along the frequency axis. The basic idea behind this approach is that the background music and vocal singing are two heterogeneous signals and the time series data of these two signals have different characteristics for each frequency bin. Figure 4-5 illustrates our approach to solve the permutation inconsistency problem.

As Figure 4-5 shows, for each frequency bin f, we have two T points complex time series output, denoted as $\mathbf{U}_i(f) = \{U_i(f,1),\cdots,U_i(f,t_s),\cdots,U_i(f,T)\}$, i=1,2, t_s denotes the time index and f is the frequency index.

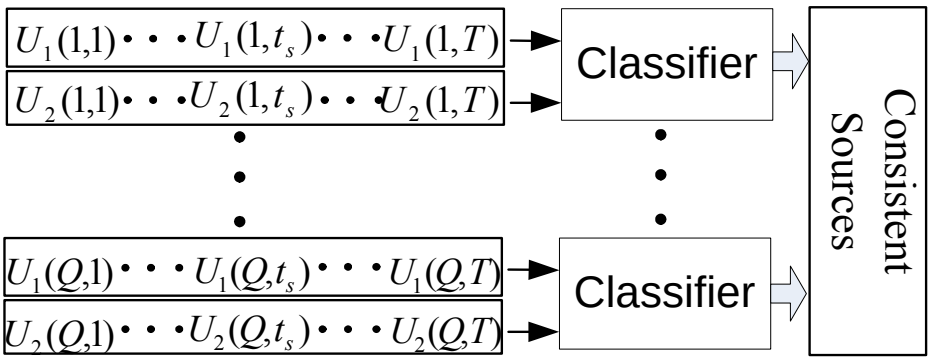


Figure 4-5: Statistical learning approach to solve the permutation inconsistency problem

Figure 4-6 depicts the magnitude of the time series of two different source signals in the same frequency bin. The horizontal axis represents the time index and vertical axis represents the magnitude. As the figure shows, the curve of the singing voice has different behavior from that of the background music. The singing voice shows some

tempo continuity, which the background music does not have. It is probably because the vocal singing is produced by vocal organ, which always stays stable for a period of time once being activated in certain frequency, while the background music consists of many music instruments and the music instruments show less stable nature than singing voice in a particular frequency bin.

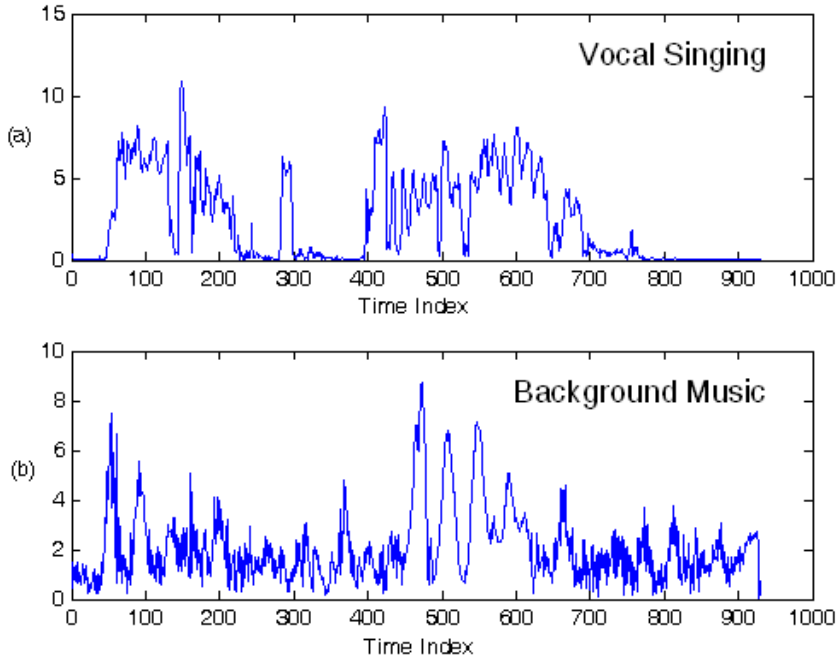


Figure 4-6: Two different output signals for a certain frequency

We first employ 13-dimentional linear prediction coefficients (LPCs) to characterize two output time series of each frequency bin with the fixed-length (i.e., 1000 time points), and followed by a Support Vector Machine (SVM) classifier to classify these two output time series of that particular frequency. SVM classifier is employed here since it is a useful statistical machine learning technique that has been successfully applied in the pattern recognition area [1]. By mapping the low dimensional feature space to the high dimensional feature space, the two-class classification problem can be made easier and more efficient. The SVM algorithm can

construct a variety of learning machines by use of different kernel functions [Appendix B.1]. We employ the radial basic function (RBF) with Gaussian kernel as the kernel function in SVM training since it is commonly used in two class classification..

After classification, the results can be denoted as $\mathbf{U}_i^P(f) = \{U_i^P(f,1),\cdots,U_i^P(f,t_s),\cdots,U_i^P(f,T)\}$, i=1,2. Along the frequency bin, $\mathbf{U}_1^P(f)$ always belongs to one particular source and $\mathbf{U}_2^P(f)$ always belongs to other source. In this way, the permutation inconsistency problem can be solved. The Figure 4-7 illustrates the results before and after the classification. Each bar represents the T-points time series of particular source in each frequency bin and the color represents the source.

As Figure 4-7(a) shows, before the classification, the time series belonging to the different sources alternatively appears in the same output entry (upper or lower of the frequency axles in the figure) along the frequency axles, due to the permutation inconsistency. In Figure 4-7(b), after the classification, the time series belonging to the same sources consistently appears in the same output entry. Therefore, to reconstruct the spectrum of one separated source for the particular time index, we can collect the value corresponding to that time index in all the time series which belong to that particular source.

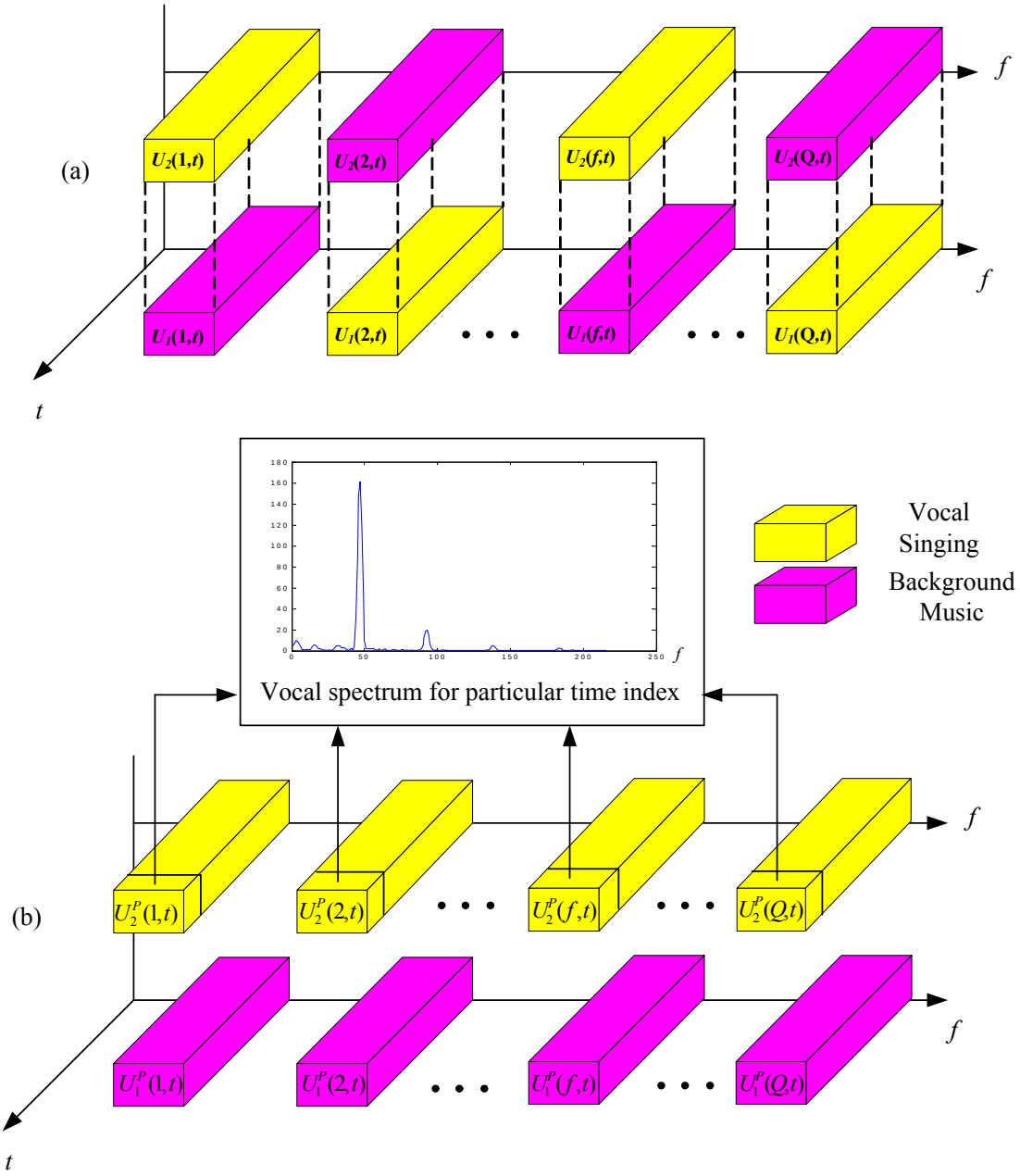


Figure 4-7: The illustration of the classification method for solving the frequency inconsistency

4.4 Query by Humming for Real World Music Database

After the vocal content has been separated from the background music, it still need to be further refined before it can be ready for the query as the monophonic representation. The major difference is that the basic unit of monophonic representation (such as MIDI) is note while the vocal content extracted from the

polyphonic music is still acoustic signal. Therefore, we need to handle the following two issues to make the QBH2 practical:

- A robust pitch detection algorithm for the separated vocal spectrum which contains interference noise and errors introduced from the separation step.
- A good note segmentation and quantization scheme for pitch contour.

Figure 4-8 illustrates the work flow of a practical QBH2 system after the vocal content has been separated.

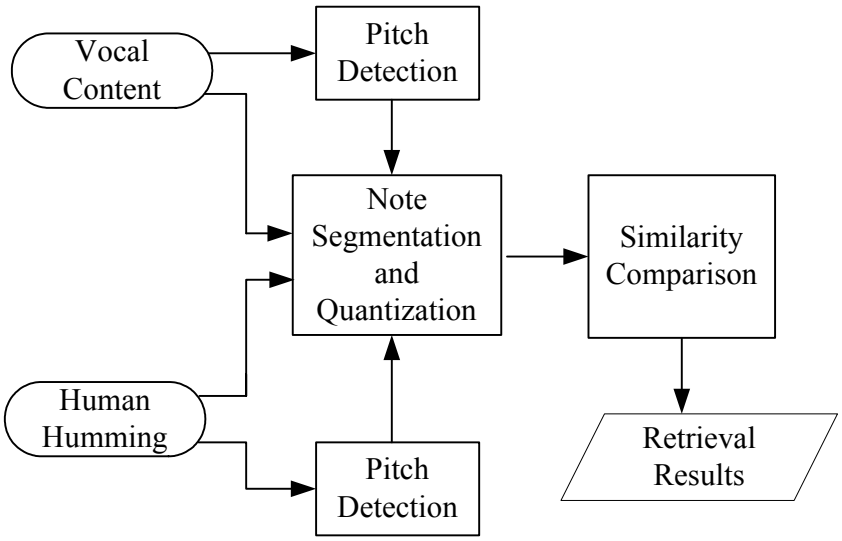


Figure 4-8: Workflow of query by humming for polyphonic music database

4.4.1 Predominant Vocal Pitch Detection

The separated vocal singing spectrum contains interference noise and errors introduced from the separation step. Among them, some of it is caused by imperfect separation, and some is caused by background music time series misclassified as singing voice time series. To correctly extract the singing pitch from the separated vocal spectrum, we have to handle these noise and errors. In our proposed approach, a smoothing function was employed to correct the misclassification errors, followed by

an algorithm to robustly locate the pitch in the vocal spectrum.

Figure 4-9(a) shows the spectrum of the vocal singing in a particular time index after the separation process. The horizontal axis represents the frequency and the vertical axis represents the magnitude. The circles denote the errors introduced by misclassification. Since the misclassification occurs only occasionally, the misclassification errors are characterized as isolated, short-term discontinuous points, which can be corrected by a smoothing function. We employ a 5-points median smoothing function followed by a 5-points Hann window linear smoothing function to correct these isolated errors. The 5 point Hann window can be defined as:

$$w(n) = 0.25 \cdot x(n-2) + 0.75 \cdot x(n-1) + x(n) + 0.75 \cdot x(n+1) + 0.25 \cdot x(n+2) \quad (4\text{-}18)$$

where $w(n)$ represents the smoothed and $x(n)$ represents the unsmoothed contour.

In Figure 4-9(b), we can easily see that the misclassification errors have been corrected after smoothing process.

After we correct the misclassification errors in the spectrum, the pitch value can be determined by the position of peaks in the spectrum. Considering the effect from local jitters and ripples introduced from separation, we propose to employ the following algorithm to robustly locate the pitch in each frame of vocal singing spectrum:

1) Identify the first 10 peaks in the spectrum with the highest magnitude, and substitute each of them with a single point in frequency, and the magnitude of each point is the height of the corresponding peak. We employ 10 peaks because in most cases, the first 10 peaks contain more than 95% of total

energy of the spectrum and are enough for pitch detection.

2) Since pitch can be measured as the greatest common divisor of the harmonics, we can estimate the pitch by compressing the 10 peak spectrums along the frequency axis with the compression factors of 2, 3, 4, etc., subsequent adding of the original and compressed spectrums, and picking up the distinct maximum. To avoid the effect of vocal formants (formants created by vocal tract of human beings often predominates the spectrum), these 10 peaks are all normalized into the unit value before spectrum compression.

3) The extracted pitch of the current frame cannot be too distant from that of adjacent frames, since the correct pitch should be stable for a period of time while the incorrect pitch does not have tempo continuity.

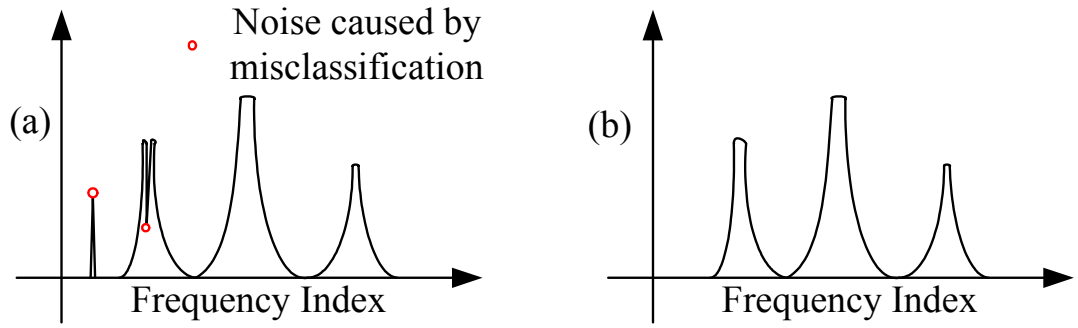


Figure 4-9: Misclassification errors correction

4.4.2 Note Segmentation and Quantization

To perform the similarity measure between the original singing voice and the varied speed of the humming voice, both note segmentation and quantization for pitch contour detection in query side and database side are necessary as the notes can be

thought as the basic units in the comparison. The note segmentation process groups contiguous pitch values which seem to be within a same note, while the note quantization process divides the pitch value of each note into discrete steps. Since the frequency transient based onset detection approach itself cannot detect the onset of gliding notes which are defined as several different music notes sung or hummed continuously without any break, we propose a note segmentation scheme based on the frequency transient onset detection, followed by a sliding window algorithm to segment the gliding notes. The segmented notes are then quantized into pitch bin according to the music scale which is logarithmic for the purpose of simulating human perception.

4.4.2.1 Note Segmentation

Once the continuous pitch values have been identified, the time locations at which a note starts (the onset time) and ends (the offset time) should be estimated. To date, no algorithm has been developed that can reliably detect the wide range of possible note onsets from different singing styles. In our specific problem (singing voice and humming), notes onset can be defined as vowel onsets. Consequently, the singing and humming are assumed to consist of short, relatively isolated syllables. Therefore, note onsets from human voice are then characterized by an abrupt rise in energy over a broad frequency spectrum and the sustained note has a relatively steady spectral shape representing the formants of the vowel used. While for note offset, although it can be identified to some extent by the fall of energy especially in higher frequency, they are

less clearly defined because the exponential decay of the amplitude of a note makes a note inaudible while it is still physically present. In light of this, we just detect the note onset as the indication of the start of a note, and we assume the note stops at the onset of the next note to avoid detecting the note offset.

The basic idea of our note onset detection scheme is that there are noticeable differences in the frequency content at note changes and we can detect the note onset by detecting the fast change transient in frequency spectrum. The proposed note onset detection scheme is shown in Figure 4-10. First, to improve the reliability of the onset detection, we divide the spectrum of input vocal signal into 2 subbands, which range from 0 Hz to 4K Hz (subband01:0~1000 Hz, subband02:1000~4000 Hz). The frequency range of subband01 is approximately corresponding to the range of the first vowel formant and the frequency range of subband02 is approximately corresponding to the range of the second and the third vowel formants. We measure the frequency transients in terms of progressive distances between the spectra in sub-band 01 to 02 using the similar method to that in [108]. After frequency transients have been detected in these two subbands, we combine them using a weighting function defined as:

$$Onset(t) = \alpha \cdot SB_1(t) + \beta \cdot SB_2(t) \tag{4-19}$$

where *Onset(t)* is the sum of onsets detected in two sub-bands and $SB_1(t)$ and $SB_2(t)$ denotes frequency transients detected in subband01 and subband02 respectively. Since the first vocal formant normally has the larger energy than other formants, to avoid

the frequency transients from subband01 dominating the output of Eq. (4-19), we set α (0.25) < β (0.75).

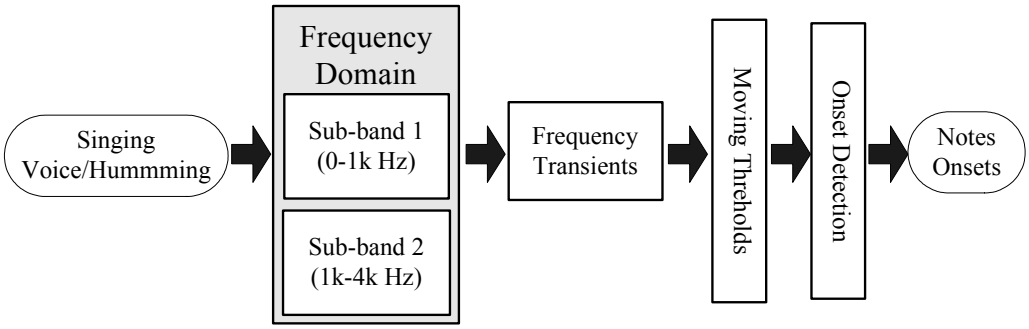


Figure 4-10: Frequency transient based onset detection scheme

After the frequency transient based note onset detection, the onsets of the isolated or individual notes (we denoted them as explicit onsets) can be successfully detected. However, for the onsets of some gliding notes (we denote them as implicit onsets), the frequency transients based onset detection algorithm cannot always correctly detect them. For gliding notes, the changes between two consecutive notes are continuous and gradual. As a result, sometimes no frequency transients can be detected in two subbands. Therefore, we propose a *sliding window algorithm* to address this problem and segment the individual notes from the gliding notes. The window is sliding along the original pitch contour between two explicit onsets detected previously and tries to find the implicit onsets. The detailed algorithm can be described as following:

1) At the beginning of each explicit onset detected, we initialize the window with the first 6 pitch values after the onset. We divide these 6 pitches into two consecutive areas, called *reservation area* and the *exploration area*, each of which contains 3 values. The reservation area maintains the pitch

of current note while the exploration area contains the incoming pitch values and always keeps fixed size (here we always keep it 3 elements).

2) The median of the pitch for each area is calculated.

If the difference of two means is smaller than a semitone (12 semitones per octave), the reservation area grows one element and the exploration area slides one element (the first elements in exploration area is included in the reservation area and the next pitch value in time line is read into the window, acting as the last element of exploration area).

If the difference of two means is larger than a semitone, the position of current boundary of two areas indicates that a new implicit onset begins. We record this position and take the median of the reservation area as the pitch value of all elements in the reservation area. At the same time, the new reservation area is the old exploration area, and the new exploration area is constructed by reading the next 3 pitch values in the time line.

Step (2) is repeated until the exploration area exceeds the next explicit onset.

It should be noted that the size of the sliding window is fixed (in our experiment it equals to 18) but the sliding window is cyclic if the exploration area exceeds the end of the sliding window.

Figure 4-11 shows the note segmentation results. The horizontal axes in all three figures represent the time index, the vertical axes in (a) and (c) represent the pitch value and the vertical axis in (b) represents the onset strength. Figure 4-11(a) is the original pitch contour detected from the polyphonic music signal after separation.

While the solid lines in Figure 4-11(b) are explicit onsets detected using the onset detection scheme. It is easy to see that the implicit onsets (appears approximately in time index 100, 300, 370 and 420) are not detected. The dot lines in Figure 4-11(b), however, indicate the implicit note onsets detected by our sliding window algorithm. An interesting point here is that an undetected explicit note onset using onset detection scheme (appears approximately in time index 220), is successfully detected using sliding window. Therefore, in some sense, the sliding window algorithm can be used to detecting the onset missed by the original frequency transient based onset detection scheme.

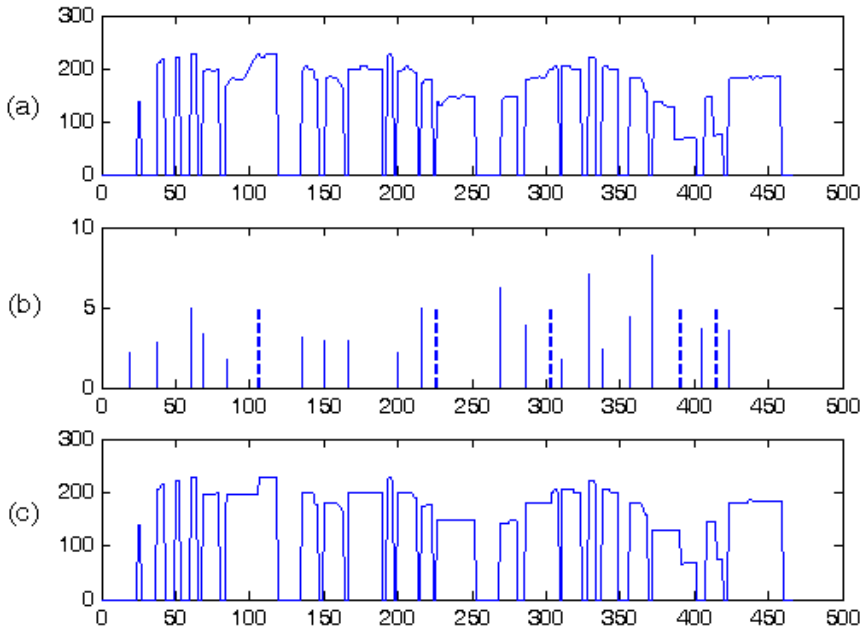


Figure 4-11: Note segmentation results

(a) Original pitch contour; (b) The onsets detected (The solid line represents the onsets detected using onset detection scheme and the dot line represents the onsets detected from gliding notes); (c) Pitch contour after sliding window algorithm.

4.4.2.2 Note Quantization

Once the notes have been segmented, we need to quantize pitch value of each note, which divides the pitch value of each note into discrete steps. This procedure is essential to measure how much the pitch of current note has changed compared with the previous and subsequent notes and is necessary for our similarity matching. Considering the fact that human perception of pitch difference is logarithmic to the frequency difference, we quantize the pitch value according to equally tempered musical scale. To give a proper pitch resolution, the quantization step is set to 1/10 semitone, which means one octave can be divided into 120 integers. For the pitch value of each note, it can be quantized as the following equation:

$$q(t) = \lfloor 120 * \log_2(p(t) / p_{\min}) + 0.5 \rfloor \quad (4\text{-}19)$$

where $p_{\min}$ is the minimum pitch value in the whole melody contour, excluding the zero value in it. In this way, the actual pitch value of each note is converted to the relative pitch value which shows relative pitch intervals among the notes. The concept of relative pitch is helpful in the following similarity matching process as absolute pitch value for each note does not make any sense due to the variety of utterances of the people with different music skills. However, the melody contour, maintained by the relative pitches, is always kept reasonably correct with the original one, for different people.

After note segmentation and quantization, the whole melody can be represented as a series of notes and the *i*-th note in the melody can be denoted as $Note_i = \{NS_i, ND_i,$

$NP_i\}$,where NS_i, ND_i and NP_i represents the starting point (Onset time), duration and quantized pitch value for note i, respectively. For the purpose of the similarity measure, the zero pitch values between two consecutive individual notes are replaced with the pitch value of previous note.

4.4.3 Similarity Measure

The aim of similarity measure is to find approximate similarities between input query melody and all the possible target melodies obtained in the database side and to select a certain number of the most similar ones in terms of similarity score. Considering the large music database, the direct comparison in note level between the query melody and the melodies in database side using traditional methods, such as Dynamic Time Warping (DTW), is computationally intensive. Therefore, the comparison between the query melody and the melodies in database side should be performed in an efficient way. In addition, considering that the tempo of human humming is different from original songs, which can be thought as a uniform stretching of the time axis, therefore, the similarity comparison should be invariant under shifting and time scaling. We propose to address the first concern by performing melody shape matching [109] to roughly filter out the most unlikely candidates in the database. Then, considering the second concern, we perform further similarity matching using Uniform Time Warping, as proposed in [110](The detailed description can be found in Appendix E), to stretch the query and all the possible candidates in a normal form to facilitate the matching process, followed by the similarity measure using Proportional

Transportation Distance (PTD) [111] (The detailed description can be found in Appendix F).

4.5 Summary

In this chapter, we propose a solution to the query by humming system for polyphonic music. The main contribution includes:

- Considering the vocal singing voice and background music as two heterogeneous signals, we present a predominant vocal content separation method for two-channel polyphonic music by employing a statistical learning based method to solve the permutation inconsistency problem in FD-ICA.
- A noise insensitive pitch detection method is specifically designed to robustly detect the vocal pitch from the noise background introduced from vocal content separation.
- To segment the discrete note in humming and singing voice, we propose a note segmentation scheme based on the frequency transient onset detection, followed by a *sliding window* algorithm to segment the glissando notes.

Experimental Results and Discussion 5

In this chapter, a series of experiments concerning the evaluation of the proposed music database structuring and retrieval algorithms are described, and the experimental results are also discussed.

5.1 Music Genre Classification Evaluation

To illustrate and evaluate our proposed musical genre classification approach, experiments are conducted for various genres of music samples.

The music dataset used in musical genre classification experiment contains 100 music samples. They are collected from music CDs and Internet, covering different genres such as Classical, Jazz, Pop and Rock. All data are sampled with 44.1 kHz sampling rate, stereo channels and 16 bits per sample.

5.1.1 Classification Results for Hierarchical Classifiers

We select 60 music samples as training data including 15 Pop songs, 15 Classical

songs, 15 Rock songs and 15 Jazz songs. Each sample is segmented into 2000 frames and the length of each frame is 20 ms. Therefore, the total number of training data is 120,000 frames. For the SVM1 which is used to classify music into Classical/Jazz and Pop/Rock, 60,000 frames including 15,000 frames of each genre are used for training. For the SVM2 which is used to classify Classical/Jazz into Classical and Jazz, 40,000 frames are used for training. Among these training frames, 10,000 frames are from SVM1 training set with 5,000 frames of Classical and Jazz respectively; the other 30,000 frames are from new training frames with 15,000 frames of Classical and Jazz respectively. For SVM3 which is used for classify Pop/Rock into Pop and Rock, 40,000 frames are used for training. The training frames selected for SVM3 is similar to those for SVM2. 10,000 frames are from SVM1 training set and 30,000 frames from new training frame. The rest 40 samples are used as a test set.

Radial basis function with c=1 is used for SVM1 and c=2 for SVM2 and SVM3 as the kernel function in SVM training and classification. After training the SVMs, we use them as the classifiers to separate Classical, Jazz, Pop and Rock frames on the test set. The test set contains 10 Classical music samples (20,000 frames), 10 Jazz music samples (20,000 frames), 10 Pop music samples (20,000 frames) and 10 Rock music samples (20,000 frames). Table 5-1 shows the number of training and test data, support vectors obtained, and test error for SVM1, SVM2 and SVM3 respectively. It can be seen that our proposed approach can achieve an average accuracy as high as 93.14% in frame based musical genre classification.

Table 5-1: SVM training and test results

	SVM1	**SVM2**	**SVM3**
Training Set	60,000	30,000	30,000
Support Vectors	4325	8327	7684
Test Set	40,000	20,000	20,000
Error Rate	6.36%	7.42%	6.79%

As the humans recognize the music genre based on the whole music pieces, we also conducted an experiment based on the music title. The whole song can be cut into several clips, and each clip lasts 30 second. Each clip will be segmented using 20ms, 50% overlapping window, and the mean feature vectors (LPCCs, Zero Crossing Rate, and MFCCs) will be calculated over the whole clip. The music genre of that music clip will be determined by these characteristic mean feature vectors. The music genre of the whole music title can be voted by the majority classes of its component clips. Table 5-2 shows the classification results based on music pieces. The column titles represent actual genre, while the row titles represent classification assigned by the system.

Table 5-2: Classification results based on music pieces

	Pop	Rock	Jazz	Classical
Pop	**90%**	0%	10%	0%
Rock	0%	**100%**	0%	0%
Jazz	10%	20%	**70%**	0%
Classical	0%	0%	0%	**100%**

From the table, we can see that Jazz music has the worst classification accuracy and is easily confused with other genres, probably due to its broad nature. Classical music seems to be easiest to classify. This makes intuitive sense because Classical music is most different from the other genres.

To further illustrate the advantage of proposed approach, we compare the

performance of proposed method and other methods including nearest neighbor (NN), and Gaussian Mixture Model (GMM). The same training set and test set are used for these methods. Table 5-3 shows the comparison result of these methods. It can be seen that our proposed method achieves a higher accuracy rate than other methods.

Table 5-3: Comparison result with other classification methods

	SVM	**NN**	**GMM**
Error Rate	6.86%	20.57%	12.31%

5.1.2 Classification Results for Unsupervised Classifier

As mentioned in section 2.3, each music piece is split into 30 second clips. Using these clips as training data, a continuous-input HMM template is created for each music piece with random initial parameters. Each state's observation distribution is modeled by a single Gaussian with 36 dimensional mean and 36 by 36 diagonal variance for MFCCs(6) and LPCCs(6) features supplemented by delta and acceleration values. Hidden state number is varied between 3, 4, 5 states. In our experiment, we found that the number of hidden states did not have dramatic impact on the system in terms of classification accuracy.

Table 5-4: 5-state HMM classification results

	Pop	Rock	Jazz	Classical
Pop	**88%**	0%	12%	0%
Rock	0%	**92%**	8%	0%
Jazz	20%	4%	**76%**	0%
Classical	0%	0%	0%	**100%**

Table 5-4 illustrates the classification results using proposed method with 5-state HMMs. The column titles represent actual genre, while the row titles represent classification assigned by the system.

It can be seen that some types of music have proven to be more difficult to classify than others. In particular, Jazz has proven to be difficult to distinguish from Pop music. It probably results from the fact that jazz music usually comprises the improvisation of the musicians, producing variations in most of the parts, which makes it similar to Pop music. Classical music has proven to be the easiest to classify. This makes intuitive sense because Classical is most different from the other genres.

For comparison, we use a fixed-length segmentation scheme with 20 ms time window and 50% overlapping to segment the music clips. As Table 5-5 shows, the average classification accuracy is 75% using the same datasets and HMM topology, which is far below that of our proposed segmentation scheme.

We also compare the performance of proposed method with other supervised learning classification method such as SVM classifier, as described in the previous sections. It was adopted because it yielded the best classification results among all supervised learning classifier. On the same dataset, as Table 5-5 shows, our proposed method is comparable to the SVM classifier. However, for SVM classifier, two problems make it inapplicable to real world applications. Firstly, from the music data point of view, SVM classifier is based on contrived taxonomies. It is not applicable to very large databases due to the ambiguities and inconsistencies in the chosen taxonomies. Secondly, from the classifier point of view, addition of new genre necessitates retraining all SVM classifiers. It is time-consuming work due to the slow training speed of SVM, especially when the genre hierarchy grows large.

Table 5-5: Comparison result

	Proposed Method	Fix-length Segmentation	SVM Classifier
Average Accuracy	89%	75%	90%

5.2 Music/ Music Video Summarization Evaluation

5.2.1 Objective Evaluation

Our aim for the music video summarization is to maximize the coverage for both music and visual content without having to sacrifice either of them. For this purpose, in the music track, we need to extract the most common and salient themes of a given music. Ideally, a music summary lasting for a longer duration should fully contain a shorter music summary.

Table 5-6 shows the music content of our test music video “Top of the world” (by Carpenter). As the table shows, section 1 and 7 (pure music section) are the Intro and Outro of the whole music track respectively, while section 2-6 belong to the principal parts of the music track. Among the principal parts, Section 2 and 5 are verses by the female singer, section 3 and 6 are chorus by male and female singers, and section 4 is the bridge portion. In this example, sections 5 and 6 are the refrains of sections 2 and 3. Music summaries extracted with respect to the changes of summary length are shown in Figure 5-1. The vertical axis represents the summary length and the horizontal axis represents the frame number (time). The bar in the figure corresponds to the frames extracted from the original music. The boundaries of each section in Table 5-6 are also labeled with the dashed line in the figure.

Table 5-6: The content of the music-“Top of the world”

Section	Range(Frame Number)	Content
1	0-20	Instrumental music as Intro
2	21-176	Verse by the female singer
3	177-227	Chorus by male and female singer
4	228-248	Instrumental music as bridge
5	249-450	Verse by the female singer
6	451-504	Chorus by male and female singer
7	505-513	Instrumental music as Outro

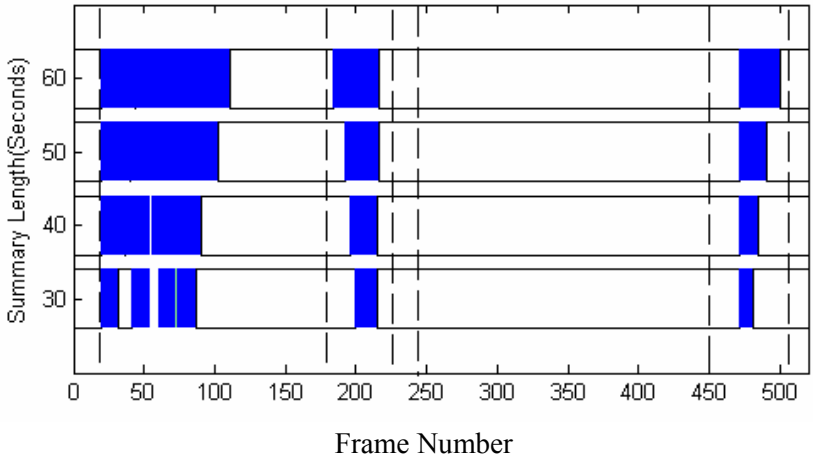


Figure 5-1: Experiment result on music video “Top of the world”

From the figure, we can see our music summarization method successfully filters the pure music portions out and performs the music summarization on the vocal music parts in section 2, 3, 5 and 6. This suggests that our proposed music classification method is efficient in separating the vocal music and pure music. The result also shows the extracted music summaries are always located at the beginning of the first verse of the music and the later parts of the two chorus portions. These excerpts were selected probably because the most salient themes of the music occurred most frequently in the memorable introductory theme and the later part of chorus. In addition, as the figure shows, a longer time length summary always completely

includes the shorter time length summaries from the same song. This indicates that our proposed method essentially captures the main theme of the music content, since to satisfy the length requirement, the shorter summary always keeps the most important part in the longer summary, and discards the less important parts. In other words, whatever the length of the summaries is, the most salient parts of the song are always included in the summaries. Another interesting observation from the figure is that all of the summaries includes two choruses in the song while for the two verses in the song, the generated summaries contains only one verse. This verifies the efficiency of our clustering algorithm as it can capture the most repeated part of the song (chorus) without missing anyone of them.

5.2.2 Subjective Evaluation

Since there is no absolute measure available to evaluate the quality of the music summary / music video summary, we employed a subjective user study to evaluate the performance of our music summarization method /music video summarization method, which is borrowed from the idea of the Questionnaire for User Interaction Satisfaction (QUIS) formulated by the Department of Psychology of University of Maryland [112]. We use following attributes to evaluate the music summary/music video summary:

a. *Clarity*: This pertains to the clearness and comprehensibility of the music video summary;

b. *Conciseness*: This pertains to the terseness of the music summary/ music video summary and how much of the music summary/music video summary captures the essence of the music/ music video;

c. *Coherence*: This pertains to the consistency and natural drift of the segments in the music summary/ music video summary.

d. *Overall Quality*: This pertains to the general perception or reaction of the users to the music summaries/music video summaries.

For the dataset, four genres of the music video are used in the test. They are Pop, Classical, Rock and Jazz. Each genre contains five music video samples. The aim of providing different music videos of different genres is to determine the effectiveness of the proposed method in creating summaries of different genres. The length of music video testing samples ranges from 2m52s to 3m33s. The length of the summary for each sample is 30s.

In this experiment, there are 20 participants with music experience, 12 males and 8 females with most of the participants being graduate students. Their ages range from 18 to 30 years old. Before the tests, the participants were asked to spend at least half an hour watching each testing sample for as many times as needed till they grasped the theme of this sample.

5.2.2.1 Subjective Evaluation of Music Summaries

We first asked the participants to evaluate our proposed music summarization scheme. We extracted the sound track of each music video and made music summary using our

proposed summarization method, which is described in section 3.2. The participants listened to the music summary one by one and rated the summary in three categories (Conciseness, Coherence, and Overall Quality) on a scale of 1-5, corresponding from the worst to best respectively. The average grade of the summaries in each genre from all subjects is the final grade of this genre. In order to make comparison and highlight the advantage of our adaptive clustering method, we also asked the participants to rate the summaries using a non-adaptive clustering method [39] in terms of same rules.

Table 5-7: Results of user evaluation of music summary

Genre	**Conciseness**		**Coherence**		**Overall Quality**	
	I	II	I	II	I	II
Pop	4.6(0.143)	3.7(0.212)	4.3(0.185)	3.6(0.283)	4.5(0.156)	3.4(0.325)
Classical	4.2(0.187)	3.0(0.243)	4.0(0.214)	3.3(0.406)	4.1(0.235)	3.1(0.386)
Rock	4.5(0.158)	3.4(0.207)	4.1(0.176)	3.5(0.277)	4.7(0.137)	3.3(0.283)
Jazz	4.3(0.202)	3.1(0.251)	4.2(0.239)	3.1(0.381)	4.2(0.279)	2.9(0.412)

I: Our proposed method

II: Non-adaptive clustering method

Table 5-7 shows the results of listening evaluation of music summaries generated by different methods. In addition, standard deviation is reported in the parentheses. From the results, we can see that our proposed method performs superior to the non-adaptive clustering method for all genres of the music testing samples in all categories in terms of high scores and low standard deviation.

5.2.2.2 Subjective Evaluation of Music Video Summaries

To evaluate our proposed music video summarization scheme, the participants were also asked to watch music video summaries generated from the testing sample using our proposed method. The participants watched the music video summary one by one

and rated the video summaries in the same manner as they rated the music summaries, except for that the video summaries were rated in the four categories (Clarity, Conciseness, Coherence, and Overall Quality) instead of three categories for music summaries. In order to make comparison, we compare the results of our proposed method with the results of music or video alone for summarization and manual summarization. On the one hand, we asked participants to rate the summaries generated automatically using music summarization only [40] and video summarization only [50], as these two methods are the most similar approaches which we can find to our proposed music summarization scheme and visual content summarization scheme, respectively. On the other hand, we also asked participants to rate the summaries manually created by the expert from EMI Singapore. In order to avoid the potential biased evaluation results, we presented the music video summaries created by different methods in a random order, and the participants did not know which technique had been used to generate each summary before they rated the summaries. Table 5-8 shows the average scores of the users' evaluation for Pop, Classical, Rock and Jazz music video summaries for various methods. In addition, standard deviation is reported in the parentheses.

From the test results, it can be seen that the summaries using the proposed method performed quite well with the score over 4 in all categories and with a low standard deviation. It also can be seen that proposed method is comparable to the manual summarization method for all genres of music video testing samples. This indicates that the proposed method is effective in realizing users' expectations.

Table 5-8 also shows that, for music video summaries generated by our proposed method, the average scores for Pop music video were generally higher than the other three genres, while the average scores for Jazz music video were generally lower than the other genres. This may be explained by the fact that usually each repetition of the main melody comes with a small variation, also depending on the genres. In most of today's Pop music, the main melody part repeats typically in the same way without major variations. In Jazz music, it usually contains the improvisation of the musicians, with variations in most of the parts. Such variations may create problems in determining the main melody part. This is consistent with earlier findings [42] suggesting that Jazz music is more difficult for clustering methods to capture the main theme of this genre compared with other music genres. It is interesting to see that our proposed method perform better than the manual method in 3 out of 4 testing genres in terms of the overall attribute. This is probably because the music expert sometimes may cut one video clip containing one of the choruses from the original video and take this clip as the music video summary, which may not give the participants a good overview of the original music video, while our proposed method maximized the coverage of the generated summaries both for music and visual contents of the original music video and is preferred by most of participants.

The music video summarization examples can be viewed at http://www.comp.nus.edu.sg/~shaoxi/Vsum/vsum1.htm

Table 5-8: Results of user evaluation of music video summary

Genre	Clarity				Conciseness			
	Proposed Method	Manual Summary	Music	Video	Proposed Method	Manual Summary	Music	Video
Pop	4.5(0.145)	4.7(0.122)	3.2(0.334)	2.9(0.125)	4.8(0.173)	4.7(0.155)	2.5(0.270)	2.3(0.234)
Classical	4.2(0.132)	4.5(0.117)	3.0(0.209)	3.0(0.268)	4.6(0.158)	4.9(0.178)	2.4(0.335)	2.8(0.189)
Rock	4.3(0.153)	4.6(0.145)	3.1(0.418)	3.2(0.189)	4.5(0.113)	4.8(0.213)	2.7(0.423)	3.0(0.282)
Jazz	4.1(0.167)	4.4(0.177)	3.0(0.264)	2.8(0.223)	4.4(0.202)	4.7(0.198)	2.9(0.372)	2.7(0.302)

Genre	Coherence				Overall Quality			
	Proposed Method	Manual Summary	Music	Video	Proposed Method	Manual Summary	Music	Video
Pop	4.4(0.175)	4.6(0.123)	3.1(0.152)	3.3(0.133)	4.5(0.113)	4.4(0.224)	2.7(0.327)	2.6(0.433)
Classical	4.0(0.108)	4.3(0.116)	3.3(0.221)	3.4(0.126)	4.6(0.150)	4.3(0.248)	2.6(0.425)	3.5(0.282)
Rock	4.2(0.134)	4.4(0.102)	3.7(0.112)	3.6(0.172)	4.3(0.182)	4.0(0.174)	2.9(0.289)	3.3(0.407)
Jazz	4.3(0.122)	4.7(0.115)	3.2(0.171)	3.5(0.163)	4.1(0.133)	4.5(0.189)	3.0(0.376)	2.4(0.269)

5.3 Query by Humming for Real World Music Database

Our experiments on query by humming for real world music database are divided into five parts. Firstly, we evaluate the performance of the SVM classifier for the alignment of frequency inconsistency. Secondly, we evaluate the performance of our proposed separation approach to the convolutive mixtures. In the third experiment, we compare the vocal pitch detected from the polyphonic recording using our proposed method with the pitch detected from the corresponding pure singing voice version. Then we test the performance of the note onset detection accuracy as our fourth experiment. Finally we evaluate the performance of our retrieval system on the polyphonic music database, by different users' inputs.

5.3.1 Performance of the Classifier

In order to evaluate the performance of SVM classifier for vocal/instrumental music classification, we conduct the following experiment. The training set contains 20 pure instrumental/vocal songs collected from the Internet and CDs, 10 are pure instrumental music and 10 are pure vocal singing (5 from female voice and 5 from male voice). The test dataset contains the 20 songs also collected from the Internet and CDs, 10 are pure instrumental song and 10 are pure vocal singing. All are Pop music sampled at 16K Hz. The training data and testing data are segmented into fixed-length and overlapping window frames (in our experiment we used 1024 samples with 50% overlapping) and the number of STFT is 2048 points after zero padding to each segmentation window. 1000 consecutive window frames are grouped as one super block and after STFT, there is one 1000-points complex time series (one time frame) in each frequency bin corresponding to one super block. Considering the frequency range of the vocal singing, which ranges from 0 to 4K Hz, we only collect the complex time frames from the frequency index 1 to 512 (512 corresponds to 4K Hz in our experiment setup), and exclude the silent time frames which can be defined as the time frames whose energy[1] is less than the predefined threshold (we experimentally set this threshold 100).

Totally 40000 time frames (half are pure vocal time frames and half are pure instrumental time frames) in the training set are collected and LPCs are calculated for

[1] The energy of the time frame **x**, **x**={$x(1),x(2),\cdots,x(512)$}, can be defined as:

$$E_n = \sum_{m=1}^{512} x^2(m)$$

each time frame to train the classifier.

After training the SVM, we test it using the time frames in testing set. The classification result on the pre-labeled test set is 95.3%. The classification result is quite good as the most of time frames can be correctly classified, and the sporadic errors can be corrected by the smoothing function introduced in section 4.4.1.

5.3.2 Vocal Content Separation Results

The success of the subsequent pitch detection for the vocal singing is dependent on the performance of the proposed vocal separation approach. In our experiment, we have to use artificial mixtures other than real world recordings because the ground truth of source signals used to create corresponding polyphonic recordings is not available (i.e. the singing version of the polyphonic music collected from Internet may not be sung by the same singer). We created 10 synthetic convolutive mixtures of one singing voice source and one corresponding background music source, each lasting 30 seconds. The sources are selected from 20 pure music/vocal songs in training set of the experiment of section 5.3.1 and four mixing filters A_{11}, A_{22}, A_{12}, A_{21} used in each mixing process are filters learnt in the polyphonic recordings. The separation results can be measured by Signal-to-Noise Ratio (SNR), which can be defined as:

$$SNR = 10 \cdot \log_{10} \frac{\sum_t \sum_f S^2(f,t)}{\sum_t \sum_f (S'(f,t) - S(f,t))^2} \quad (5\text{-}1)$$

where $S(f,t)$ denotes the discrete spectrum representation of pure singing voice, and $S'(f,t)$ denotes the vocal content spectrum obtained using our proposed method. To make comparison, we also employ the method proposed in [81] and [83] using the

same dataset. The average SNR for each method to separate the vocal content from these 10 mixed songs is reported in Table 5-9. In addition, the standard deviation for each method is also included in the table. As the table shows, the high SNR and low standard deviation represents the effective of our proposed separation scheme. In order to highlight the contribution of SVM classifier, we also compare with the performance of FD-ICA algorithm without employing SVM classifier to align the permutation inconsistency. From the result of experiment, we can see that, without any permutation alignment, the average SNR is only 2.51 dB and is much worse than any methods with permutation alignment.

Table 5-9: Vocal separation performance of different approaches

	Average SNR(dB)	**Standard Deviation**
Proposed method	10.57	1.4896
Smaragdis's method	7.96	1.6153
Mitianoudis's method	8.74	1.8332
FD-ICA without SVM	2.51	3.6358

5.3.3 Pitch Detection Experiment Results

In this experiment, the test dataset contains 40 polyphonic music excerpts extracted from 20 polyphonic songs from Internet and CDs. All are Pop songs and sampled at 16 kHz. We also collected the pure singing version of corresponding music excerpt from the Internet as the ground truth, due to the fact that although the polyphonic version and pure singing version are sung by different singers, the melody contours of singing in these two versions are similar.

In order to measure the similarity of these two pitch contours, we first convert the pitch value into music cents according to its frequency value, and the fact that the

smallest interval in western music is 100 cents (one semitone) is used to group a sequence of samples into one note. The two note contours are aligned in time manually and we denote a character “U” at the current note if the note is higher than previous one and a “D” if the note is lower than previous one. The matching accuracy can be defined as the number of matching notes divided by total number of notes in comparison.

In our experiments, for the 40 music excerpts, after vocal content separation process, the average matching accuracy of our proposed pitch detection approach is 81.4%. To make a comparison, we also employ an audio tool called Praat (which is available at http://www.fon.hum.uva.nl/praat/) to detect vocal pitch after separation. The average matching accuracy of that method is 74.1%, which is relatively lower than our proposed method. It is probably due to the fact that the pitch detection algorithm provided by Praat doesn’t deal with the misclassification error in vocal content separation process before it detects the pitch values. To verify this, we also use Praat to detect the pitch after misclassification error correction introduced in section 4.4.1.This time, the average matching accuracy is 80.7%, and is comparable to our proposed approach. Although our proposed pitch detection approach can not correctly detect the relative pitch at all points, our proposed pitch detection approach is still quite good since pitch contour can be represented properly with such average accuracy, and for the retrieval task, the individual pitch is less important than the pitch contour.

5.3.4 Note Onset Detection Accuracy

The proposed note onset detection method can be tested on singing voice segregated from polyphonic music and humming of human voice. The test set here are 10 pieces of pure singing voice (from 10 pure vocal songs in training set in section 5.3.1) and 10 pieces of humming collected from 3 males and 2 females, and each lasting for about 20 seconds. For each singing or humming clip, the onset points were manually labeled beforehand with the aid of corresponding music scores, which help accurately locate the onset in the singing or humming voices.

An onset is considered accurately detected if it falls within 100ms window of pre-labeled onset position, and an onset is undetected if there is no onset detected using our algorithm within 100ms window around pre-labeled onset position. An onset is considered falsely detected if it falls outside 100 ms window around pre-labeled onset position. Table 5-10 shows the onset detection results both for the singing and humming (Using 'La') from different human voices using our proposed approach and the frequency transient approach similar to the one proposed in [108]. As Table 5-10 shows, the accuracy of proposed onset detection (92.95% for singing and 91.69% for humming) is much higher than the previous frequency transient based approach (80.22% for singing and 62.46% for humming), as most of the implicit onsets of glissando notes in singing voice and humming are successfully detected in our approach. It can also be seen that the performance on humming voice increases more significantly than on singing after using our proposed method. This is probably due to the fact that people are prone to humming more glissando notes than when they

are singing.

Table 5-10: Onset detection results

	Total	Frequency Transient Method		Our Method	
		Correct	False Alarm	Correct	False Alarm
Singing	369	296	17	343	25
Humming	325	203	21	298	32

5.3.5 Performance of the Retrieval System

In query by humming system, people are used to humming a tune belonging to the music sections containing singing voice such as chorus and verse due to the fact that the vocal content section of a song is easier to remember than the non-vocal sections for the human beings. Therefore, music semantic region detection is important to filter out some non-vocal sections in a song such as Introduction (Intro), Bridge, Instrumental and Ending (Outro), and we only need to compare the input query with the music sections containing singing voices such as chorus and verse. Here, we employ the music semantic region detection algorithm proposed in [32] to automatically detect the chorus and verse.

To evaluate the performance of our proposed retrieval system in real world sound recordings in the database side, we collected 100 polyphonic songs extracted from CDs. All are English Pop songs. After semantic region detection as proposed in [32], we totally have 771 vocal instrumental mixture segments (303 verses and 468 choruses). As for the query side, we ask 10 people, 5 females and 5 males, to hum for the system. Each person hummed the melodies of 4 different songs through

microphone (2 melodies belong to the chorus parts of two different songs and 2 melodies belong to the verse parts of another two different songs). The retrieval results is returned to the users as a ranked list ordered from high to low in terms of similarity measure with the input query. The retrieval accuracy can be defined as the number of target songs falls in top-n list divided by the number of total queries. For example, in our experiment the total number of queries is 40. For these 40 queries, the retrieval system will return 40 rank lists as the retrieval results, and among these rank lists, suppose the number of target songs appearing as the first item of the list is 10, and then the retrieval accuracy for top-1 list is 25%. Table 5-11 shows the average retrieval accuracy of our proposed retrieval system with different rank list numbers.

As Table 5-11 shows, the accuracy of target songs appearing in the top 1 rank list is low, compared with the accuracy obtained on the MIDI database (usually above 50% for top 1 rank) [109], but our proposed method performs on the real world polyphonic music signals. Unlike MIDI files, we cannot perfectly obtain the melody information from polyphonic music signals. As the ranked list size increases, the retrieval accuracy of our retrieval system is more and more close to that of retrieval system on MIDI database. This is because although our proposed algorithm cannot perfectly obtain the melody information from polyphonic music signals, it still can reasonably approximate the correct melody. Therefore, when we relax the restriction by increasing the rank list number, the target songs can then be found. This point is a good characteristic for the application of our proposed system, as the people in the query side normally have end facilities such as hand phone or PDA, high accuracy for

top 10 rank and low accuracy for 1 or 2 rank is still acceptable since they can select the desired song in the screen which can easily display the titles of 10 songs.

Table 5-11: Retrieval accuracy for our proposed method

Rank list Number	No. of Target Songs Falling in	Accuracy
1	15	37.5%
2	22	55.0%
5	29	72.5%
10	34	85.0%

In addition, among the 6 target songs which are not included in the top 10 list, we also found some of them had been filtered out by the previous melody matching algorithm due to the incorrect music semantic region detection, i.e. the semantic region detection algorithm segments the incomplete choruses or verses, while the people accidentally hum the tunes on these incomplete choruses or verses part. Since the corresponding melody is missed in the target chorus or verse, the target chorus or verse will be rejected by our current melody shape matching. To remove the effect of automatic music semantic region detection to the retrieval accuracy, we also conducted the experiment on the manually labeled music semantic region database. Table 5-12 shows the retrieval accuracy on this experiment. From the table, we can find that with manually labeled music semantic region, the retrieval accuracy increase slightly for all rank list number except the rank list number 1, as the incorrectly filtered out target semantic region in automatic approach can be avoid. Although the music semantic region segmentation and melody shape matching slightly decrease the accuracy of the system, these two steps are still indispensable because these two steps significantly reduce the searching space and the retrieval process will be very

complex or even impossible without these two steps.

Table 5-12: Retrieval accuracy for manually labeled music semantic region

Rank list Number	No. Of Target Songs Falling in	Accuracy
1	15	37.5%
2	23	57.5%
5	31	77.5%
10	36	90.0%

Our current music retrieval system will work well for songs which have only two kind of heterogeneous signals mixed together. However, it will not work well if more than one singer are singing simultaneously in the song, which may happen in the chorus. Under this circumstance, our separation approach may fail as the separated vocal contents are still mixed. Another limitation for our system is that it cannot be applicable to the songs containing heavy metal instrumental music, because such kind of music will destroy the assumption that background music has super-Gaussian distribution.

5.4 Summary

In this chapter, a series of experimental results concerning the evaluation of the proposed music database structuring and retrieval algorithms have been described. For the music genre classification, the result of our proposed multi-layer SVM classifier achieves a higher accuracy rate than other prescriptive approaches, and the result of our proposed unsupervised learning approach for music genre classification also achieves a promising accuracy. For music/music video summarization, objective evaluation and subjective evaluation indicates that music or music video summaries

generated using the proposed method is effective in helping realize users' expectations. In addition, several experiments related to humming based music information retrieval for the real world music database were described in this chapter. It was shown that the humming based music information retrieval for the real world music database can still achieve relatively high retrieval accuracy using digital signal processing method, when combined with the machine learning approach.

Conclusions 6

In this thesis, several issues in real world digital music database management were tackled. These issues include music summarization, music genre classification and music retrieval by human humming. We have shown that these problems can be solved by digital signal processing method, combined with the various machine learning approaches. Music summarization and music genre classification are categorized as the middle level music understanding applications, while music retrieval is categorized as the high level music interactive application. In the context of music perception architecture, the two middle level music understanding applications provide the structure information of music database and individual songs respectively and address the issues of how to organize the music database. In this way, interaction with the music database can be made effective and efficient.

6.1 Summary of the Contributions

For music genre classification, we presented two approaches for automatically classifying music genres, one is based on supervised learning and the other is based

on unsupervised learning. Both approaches extract the genre information from the music content itself rather than the metadata annotation. For the supervised learning approach, we propose a multi-layer SVM classifier to hierarchically distinguish music genres. In this approach, the music classification problem can be solved by multi-layer classification scheme, in which the classifiers in different layer perform binary classification and use level-dependent and genre-specific features. The advantage of this method is that each classifier deals with an easier separable problem and we can use an independently optimized feature set at each step. The experimental results demonstrate superior performance compared to the existing supervised learning approach. The disadvantage of this approach is also obvious. First, the hierarchical system is difficult to expand both in width (new root genres) and depth (new subgenres), since adding new music genres is equivalent to adding a new classification tree or new leaves to an existing tree, and the optimized feature set of the related level would have to be re-selected. Secondly, as already mentioned, the supervised approach depends on the contrived taxonomy, which currently is ambiguous and inconsistent. To avoid the ambiguities and inconsistencies caused by contrived taxonomy given a priori, we proposed an unsupervised music genre classification method, which takes the advantage of the similarity measure to organize the music collection with clusters of similar songs. In this way, the ambiguities and inconsistencies in built-in taxonomy for supervised approach can be partially avoided. The experimental results show that the accuracy of this unsupervised approach is comparable to that of the previous supervised approaches in a small taxonomy.

However, the major drawback of our unsupervised method maybe that the obtained clusters are not labeled. To recognize the genre information of songs in the cluster, human intervention is still necessary.

For music summarization, we have proposed an approach which extracts the most salient part of music based on adaptive clustering, with the help of music structure analysis. Prior research has addressed the problem of finding the most salient frames or the segments of a song. However, they all fail to distinguish between the pure instrumental music and vocal music during the process of music summary generation. As a result, a summarized segment may contain unwanted pure instrumental music portions. This is definitely not desirable for the purpose of understanding music content, since according to music theory, the most distinctive or representative music themes should repetitively occur in the vocal part of a music work. The contributions of our approach are multifold. First, we summarize a song by differentiating the roles of the different parts in the song. Secondly, we employ an adaptive clustering approach to find the main theme of the music. Another contribution of this research has been the development of the performance measurement method for evaluating music summaries. Since there is no ground truth to evaluate whether the extracted highlight is able to represent the most interesting and salient parts of a given music content, we have employed an evaluation system which employs different attributes related to the users' perception of the music summaries, borrowing the idea from the Questionnaire for User Interaction Satisfaction (QUIS) study formulated by the Department of Psychology of University of Maryland. The subjective evaluation

results show that our proposed method performs better performance than the previous methods [39][40][41].

As an extension of music summarization, a music video summarization scheme is also proposed in this thesis. In our proposed music video summarization approach, we first generated the summary for the music track and the visual track separately, and then a visual and audio alignment algorithm was proposed to generate the final summary for music videos. The proposed alignment algorithm maximizes the coverage of important audio segments along with important video segments.

Query by humming for real world music database (QBH2) is an interactive application, which belongs to the top level in the human music perception architecture. There did exist lot of work on query by humming for monophonic music database, which is relatively simpler than QBH2, since the standard music presentation (sequence of the notes) can be easily obtained from monophonic than from polyphonic music database. In QBH2 system, such kind of basic representation for music is difficult to obtain due to two difficulties. First, separating one monophonic representation from the polyphonic music is difficult, as the individual monophonic representations in polyphonic music interfere with each other both in time domain and frequency domain. Second, after the monophonic representation has been extracted out, it still needs some efforts to convert such monophonic representation to the standard format of the music representation, as the extracted monophonic representation is still acoustic signal, not note sequences. In our proposed QBH2 system, we tacked the first difficulty using FD-ICA approach, as the instantaneous

mixture separation approach is not capable of finding the correct solution to the real world convolutive mixtures. The most notable unsolved problem in FD-ICA approach is *permutation inconsistency* problem, which significantly degrades the separation performance. In our proposed approach, we exploit the heterogeneous properties of the vocal singing and background music signals, and solve the permutation inconsistency problem by employing a statistical learning based approach. The comparison studies show our proposed separation scheme achieves high SNR and low standard deviation, compared to previous approaches in FD-ICA. To tackle the second difficulty in the QBH2 system, we propose a note segmentation scheme based on the frequency transient onset detection, followed by a *sliding window* algorithm to segment the explicit and implicit notes from the monophonic representation. After these two steps, the QBH2 problem can be converted to QBH1 problem, and all the current similarity measuring approaches in QBH1 can be applied to match the input query with the melody contour in the music database side.

6.2 Future Work

Content based music management in the real world database is still a new area that has not been well explored, and a lot of interesting directions need to be investigated in the future. Some of these directions are obvious extensions of our work in this thesis and others appear unrelated.

In this thesis, our unsupervised method for music genre classification considered only classifying music into broad and significantly different categories, and it cannot

classify the music genres that have minor differences. On the other hand, the supervised learning classification system can distinguish such trivial differences between genres better than the unsupervised one, but it assumes a pre-existing taxonomy that the system can learn. Therefore**,** one obvious future direction is to scale-up the unsupervised classification, combined with the supervised approach, to real world large scale database. For example, our proposed unsupervised method could be employed to do broad initial classification with significantly different categories, and the supervised approach could then be employed to classify the fine subcategories. Some efforts are still needed to further explore the possibility of combining unsupervised and supervised approach, to utilize the strengths of both. Of course, the success of large scale music genre classification is greatly dependent on the new genre taxonomy in which consistency is maintained. In addition, such new taxonomy should support *evolvability*, i.e. be able to cope with new emerging genres. However, to define a new taxonomy is easy, and to make everyone agree on such a standardized system would be very difficult.

For music summarization, our proposed method worked well only for music genres that have constrained music repetition patterns. Therefore, further work will be needed to improve the accuracy of the summarization result for other music genres that have a free music style. To achieve this goal, more music features that can be used to characterize music content are needed to be further explored. For example, accurate rhythm features will be significantly helpful in the music summarization, since rhythm information can help identify the boundaries of music phrases. Existing

beat-tracking systems are useful in acquiring rhythmic features. As a result, the incomplete music phrases will not be included in the music summary. However, many existing beat-tracking systems provide only an estimate of the main beat and the strength of the main beat, and cannot accurately describe the rhythmic contents of the music. In addition, more domain-related music knowledge should also be taken into consideration when generating music summary.

In music video summarization, our proposed approach works well only for music video in polyphonic structure with weak synchronization of the audio and visual content. For a music video in homophonic structure with strong synchronization of the audio and visual content, this approach may sacrifice the synchronization when generating the summary. In future work, we will explore effective methods to create the summary for music videos having homophonic structure. One possibility is to detect the chorus of the music, shot boundaries and the most repeated lyrics from low-level audio/visual features and align the boundaries of the chorus, shots and lyrics based on music knowledge. We believe that the combination of complementary strengths of low-level features and high-level music knowledge is necessary to analyze and summarize the music video content.

In query by humming for real world music database system, the most difficult problem is to separate one individual source from the polyphonic music. We employ the FD-ICA approach as this approach is probably the most practical approach for separating the real world convolutive mixtures. We proposed to solve the *permutation inconsistency* problem in FD-ICA by a machine learning approach. This is based on

the assumption that there are only two kind of heterogeneous signals mixed together. However, this assumption may not hold as the more than one singer is singing at the same time in the song, which may happen in the chorus. Under this circumstance, our separation approach may fail as the separated vocal contents are still mixed. One of the direct future directions is to investigate on this problem. In addition, our proposed approach works for two channel polyphonic music, and there are still lots of mono channel polyphonic music. The query by humming system built on the mono channel will be a challenging future direction. Some work has reported on tackling this issue[113][114], but it make many assumptions and require information about the source signals, which makes it more difficult to apply to a real world problem than the conventional Blind Source Separation techniques. Furthermore, another interesting direction for the future work is to introduce some users' relevance feedback to improve the retrieval accuracy of our system, as the relevance feedback mechanism has been proven to be an efficient solution for improving the retrieval accuracy in content based image retrieval [115][116].

Generally speaking, music management in the real world database is far from mature, and there is a gap between the high level applications and low level representation of the music. Large parts of more interesting tasks, such as generic features, automatic polyphonic transcription and instrumental tracking, fall into this gap. To fill the gap with completely automatic systems, on the one hand, it might be necessary to find digital signal processing algorithms for accurately and faithfully representing music content. On the other hand, machine learning approaches for better

understanding the music content are also indispensable. With the maturation of techniques in music representation, understanding and interactions, it is possible to bring fundamental changes to the way people access and manage the ever increasing music databases.

Appendix A. Music Features

A.1 Beat Spectrum

The beat spectrum can be calculated from the music using three principal steps [28].

First, the music is parameterized using a spectrum or other representation. This results in a sequence of feature vectors $\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_i, \cdots, \boldsymbol{V}_n$.

Second, a distance measure is used to calculate the similarity between all pair-wise combinations of feature vectors. The obtained similarity is embedded into a two dimensional representation called similarity matrix **S**. The similarity between vectors $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$ can be defined as:

$$D_C(i,j) = \frac{\mathbf{V}_i \bullet \mathbf{V}_j}{\| \mathbf{V}_i \| \| \mathbf{V}_j \|} \tag{A-1}$$

Finally, the beat spectrum can be obtained from finding periodicities in the similarity matrix, using diagonal sums or auto-correlation. Both the periodicity and relative strength of rhythmic structure can be derived from the similarity matrix **S**. We call a measure of self-similarity as a function of the lag the *beat spectrum B(l)*.Peaks in the beat spectrum correspond to repetitions in the audio. A simple estimate of the

beat spectrum can be found by summing **S** along the diagonal as follows:

$$B(l) = \sum_{k \subseteq R} \mathbf{S}(k, k+l) \tag{A-2}$$

$B(0)$ is simply the sum along the main diagonal over some continuous rang R, $B(1)$ is the sum along the first superdiagonal,and so forth.

A.2 Linear Prediction Coefficients(LPCs)

The basic idea behind linear predictive analysis is that a specific time series sample at the current time can be approximated as a linear combination of past samples. Through minimizing the sum of squared differences (over a finite interval) between the past samples and linear predicted values, a unique set of parameters or predictor coefficients can be determined. These coefficients form the basis for linear predictive analysis of real valued time series. For a time series $s(k)$,k=1,…,N, a linear predictor with prediction coefficients, a_k is defined as a system whose output is:

$$\bar{s}(n) = \sum_{j=1}^{p} a_k s(n-j) \tag{A-3}$$

Where p is number of samples used in estimation.

The prediction error in the time index n, $e(n)$ can be defined as:

$$e(n) = s(n) - \bar{s}(n) \tag{A-4}$$

The criteria is to make the average prediction error minimum.The average prediction error can be defined as:

$$E_n = \sum_{m=1}^{N} e_n^2(m) \tag{A-5}$$

A.3 LPC derived Cepstral coefficients (LPCCs)

An alternative feature for LPC coefficients is LPC derived Cepstral coefficients, which can be computed simply as:

$$c_n = -a_n + \frac{1}{n}\sum_{i=1}^{n-1}(n-i)\cdot a_i \cdot c_{n-i} \qquad \text{(A-6)}$$

where a_i (i=1,…,n) is the LPC coefficients.

A.4 Zero Crossing Rates

In the context of discrete-time signals, a zero-crossing refers to two successive samples having different algebraic signs. The N-length short time zero crossing rates are defined as:

$$Z_s(m) = \frac{1}{N}\sum_{n=m-N+1}^{m}\frac{\left|\mathrm{sgn}\{s(n)\}-\mathrm{sgn}\{s(n-1)\}\right|}{2}w(m-n) \qquad \text{(A-7)}$$

Where $w(m)$ is a rectangle window which has N samples.

A.5 Mel-Frequency Cepstral Coefficients (MFCCs)

Mel-Frequency Cepstral Coefficients (MFCCs) [12] are a common feature frond-end that is used in many speech recognition systems and they are employed to model the human perception to the audio signals. More specifically, MFCCs can be calculated as following:

1. The short-time slice of audio data to be processed is segmented with a hamming window
2. The magnitude of the Discrete Fourier Transform is computed using the FFT

algorithm.

3. The FFT power coefficients are filtered by a triangular band-pass filter bank. The filter bank consists of K=19 triangular filters. Denoting the output of k-th filter bank by S_k (k=1,2,…K), the MFCCs can be calculated as:

$$c_n = \sqrt{\frac{2}{K}} \sum_{k=1}^{K} (\log S_k) \cos[n(k-0.5)\pi / K] \quad n = 1,2,...,L \qquad \text{(A-8)}$$

where L is the number of cepstral coefficients.

Appendix B. Machine Learning

B.1 Support Vector Machine

Support vector machine (SVM) learning is a useful statistic machine learning technique that has been successfully applied in the pattern recognition area [1][117].

Suppose we are given a set of training data (x_1,x_2,…,x_n) and their class labels (y_1, y_2,... y_n) , where $x_i \in R^n$ and $y_i \in \{+1,-1\}$. and we want to separate the training data into two classes. If the data are linearly non-separable but nonlinearly separable, the non-linear SVM classifier will be applied.

The basic idea is to transform input vectors into a high dimensional feature space using non-linear transformation Φ, and then to do a linear separation in feature space.

To construct a non-linear SVM classifier, inner product $<x,y>$ is replaced by a kernel function $K(x,y)$.

$$f(x)=sgn(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b) \tag{B-1}$$

The SVM algorithm can construct a variety of learning machines by use of different kernel functions. Three kinds of kernel functions are usually used. They are:

1) Polynomial kernel of degree d

$$K(x, y) = (< x, y > +1)^d \tag{B-2}$$

2) Radial basis function with Gaussian kernel of width C >0

$$K(x, y) = \exp(-|x - y|^2 / c) \tag{B-3}$$

3) Neural networks with tanh activation function

$$K(x, y) = \tanh(k < x, y > + \mu) \tag{B-4}$$

Where the parameters k and μ are the gain and shift.

B.2 Comparison of Two Hiddem Markov Models

In this section, we will describe the similarity measurement of two Hidden Markov Models.

A Hidden Markov Model has several components. It can be completely defined by the number of hidden states, a static state transition probability distribution **A**, the observation symbol probability distribution **B** and the initial state distribution $\boldsymbol{\pi}$. We can define one HMM model as $\lambda = \{\mathbf{A}, \mathbf{B}, \boldsymbol{\pi}\}$.

The training process is to use algorithms such as the Baum-Welch algorithm [34] to learn the parameters of HMM from the training samples, typically a sequence of observations. By Baum-Welch algorithm, it is possible to train the HMM by adjusting the weights of the transitions, the initial state distribution and the observation symbol probability in each state, to better model the relationship of the actual training samples.

Considering the case of two models, $\lambda_1 = \{\mathbf{A}_1, \mathbf{B}_1, \boldsymbol{\pi}_1\}$, $\lambda_2 = \{\mathbf{A}_2, \mathbf{B}_2, \boldsymbol{\pi}_2\}$,

We can generalize the concept of model distance by defining a distance measure $D(\lambda_1, \lambda_2)$, between two Markov models, λ_1 and λ_2, as:

$$D(\lambda_1, \lambda_2) = \frac{1}{T}[\log P(\mathbf{O}^{(2)} \mid \lambda_1) - \log P(\mathbf{O}^{(2)} \mid \lambda_2)] \tag{B-5}$$

Where $\mathbf{O}^{(2)}=(\mathbf{o}_1\mathbf{o}_2\mathbf{o}_3\cdots\mathbf{o}_T)$ is a sequence of observations generated by model λ_2. Basically, the Eq.(B-5) is a measure of how well model λ_1 matches observations generated by model λ_2, relatively to how well model λ_2 matches observations generated by itself.

One of the problems with the distance measure of Eq.(B-5) is that it is nonsymmetrical. Hence a natural expression of this measure is the summarized version:

$$D_s(\lambda_1,\lambda_2)=\frac{D(\lambda_1,\lambda_2)+D(\lambda_2,\lambda_1)}{2} \qquad \text{(B-6)}$$

Appendix C .Information Theory

Information is closely related to randomness or suprisal of an outcome.In this appendix, we will present part of information related to our thesis.Interested readers can refer to [117] for a complete intruction.

C.1 The Definition of the Entropy

The entropy of one discrete random variable can be defined as:

$$H(X) \equiv -\sum_{x \in \aleph} P(x) \log P(x) \tag{C-1}$$

Where X is the random variable, $P(x)$ is the probility of random variable X takes the certain value in a alphbet set, and $\aleph$ is the alphbet set that X belongs to.

For the continous case, the entropy of continous random variable is called differential entropy,which can be defined as:

$$h(x) \equiv -\int_{-\infty}^{+\infty} f_X(x) \log f_X(x) dx \tag{C-2}$$

Where $f_X(x)$ is the PDF of continous random variable x.

The entropy is the fundamental measure of information theroy. It is a very broad concept and it is used to measure the uncertainty of a random variable.

C.2 The Definition of the Joint Entropy

The joint entroy of two random variables X and Y is defined as:

$$H(X,Y) \equiv -\sum_{x\in\aleph, y\in\Im} P(x,y)\log P(x,y) \tag{C-3}$$

Where $P(x,y)$ is the joint probility of random variable X *and* Y.

For the continous case,

$$H(X,Y) \equiv -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_{X,Y}(x,y)\log f_{X,Y}(x,y)dxdy \tag{C-4}$$

The joint entropy is a measure of overall incertainty of a set of variables.

C.3 The Definition of the Conditional Entropy

The conditional entroy of two random variables X and Y is defined as:

$$H(X \mid Y) \equiv -\sum_{x\in\aleph, y\in\Im} P(x,y)\log P(x \mid y) \tag{C-5}$$

For the countinous case:

$$H(X \mid Y) \equiv -\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_{X,Y}(x,y)\log f_{X|Y}(x \mid y)dxdy \tag{C-6}$$

The conditional entroy is a measure of uncetainty of X given certainty of Y.

C.4 Kullback-Leibler (K-L) Divergence

The Kullback-Leibler (K-L) divergence is also-called relative entropy. It measures the difference between two probability distribution $P(x)$ and $Q(x)$. The definitions are:

$$D_{P\|Q} \equiv \sum_{x} P(x)\log\frac{P(x)}{Q(x)} \tag{C-7}$$

For the continuous case:

$$D_{f\|g} \equiv \int_{-\infty}^{+\infty} f(x)\log\frac{f(x)}{g(x)}dx \tag{C-8}$$

Where $f(\cdot)$ and $g(\cdot)$ are the two PDFs of a continuous random variables, respectively.

C.5 Mutual Information

The mutual information $I(X, Y)$ is a special form of the relative entropy, and the mutual information between two random variables X and Y from set $\aleph$ and $\Im$ is given by:

$$I(X,Y) \equiv D_{P(x,y)\|P(x)P(y)} = \sum_{x\in\aleph}\sum_{y\in\Im} P(x,y)\log\frac{p(x,y)}{P(x)P(y)} \tag{C-9}$$

For the continuous case:

$$I(X,Y) \equiv \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f_{X,Y}(x,y)\log\frac{f_{X,Y}(x,y)}{f(x)f(y)}dxdy \tag{C-10}$$

The relationship between the marginal entropy $H(X)$ and $H(Y)$, joint entropy $H(X,Y)$, conditional entropy $H(X|Y)$,$H(Y|X)$ and mutual information $I(X,Y)$ can be shown in the Figure C-1.

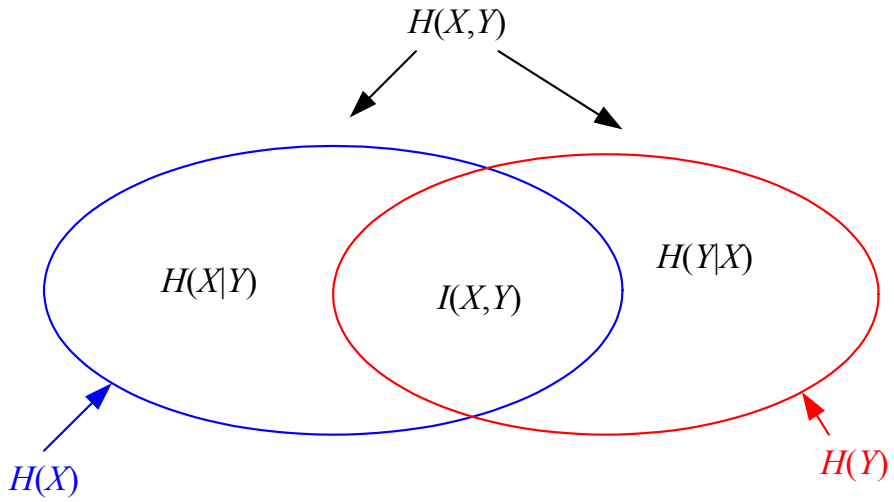


Figure C-1: The relationship between marginal entropy joint entropy, conditional entropy and mutual information

C.6 Maximum Entropy Theory

Under certain constrains, it is possible to find a random variable whose PDF has the maximal entropy. Here, the maximum entropy of a distribution is derived for an amplitude bounded random variable.

Theorem C-1(Maximum entropy of an amplitude bounded random variable):

The entropy of an amplitude bounded random variable X is $H(X) \le \log |N_X|$, *where* N_X *denotes the number of elements in the range of X, with equality if and only if X has a uniform distribution over* N_X.

Proof C-1: Let $q(x) = \frac{1}{|N_X|}$ be the uniform PDF over N_X and let $p(x)$ be the PDF of X, then:

$$D_{p\|q} = \sum p(x) \log \frac{p(x)}{q(x)} = \sum p(x) \log |N_X| - \sum p(x) \log \frac{1}{p(x)}$$

$$= \log |N_X| - H(X)$$

Since the relative entropy is non-negative, it follows that:

$$0 \le D(p \,\|\, q) = \log |N_X| - H(X)$$

And therefore,

$$H(X) \le \log |N_X|$$

Hence, the uniform distribution has the highest entropy when X is of given amplitude range.

Appendix D. Derivation of ICA for Instantaneous Mixtures

This appendix is divided in to two parts. The first part shows the derivation that Informax approach to ICA learning rule, and the second part shows the derivation that Minimizing Kullback-Leibler divergence approach to ICA learning rule. The unmixing structure that was used here is the one in Figure 4-2.

D.1 Informax Approach

Bell [78] starts by making an output entropy maximization learning rule for a 1 input by 1 output problem, and then generalizing the learning rule for the 2 by 2 case later.

Assume an input random variable X, and an output $Y = g(w \cdot X)$, where w is an arbitrary weight variable, and $g(\cdot)$ is a non-linear function. Our goal is to find the value of w which maximizes the entropy of Y.

The entropy of Y can be defined as:

$$H(Y) = -\int_{-\infty}^{+\infty} f_Y(y) \cdot \log f_Y(y) dy \qquad \text{(D-1)}$$

Where $f_Y(Y)$ is the PDF of random variable Y, which can be computed given the PDF of X from:[118]

$$f_Y(y) = \frac{f_X(x)}{\left|\frac{\partial Y}{\partial X}\right|} \tag{D-2}$$

Substitute the (D-2) into (D-1), we have:

$$H(Y) = \int_{-\infty}^{+\infty} f_Y(y) \cdot \log\left|\frac{\partial Y}{\partial X}\right| dy - \int_{-\infty}^{+\infty} f_Y(y) \log f_X(x) dy \tag{D-3}$$

The second term on the right may be considered to be unaffected by alterations in a parameter w determining $g(x)$. Therefore, in order to maximize the entropy of Y by changing w, we need only concentrate on maximizing the first term. A stochastic gradient learning rule for

$$\Delta w \propto \frac{\partial H}{\partial w} = \frac{\partial}{\partial w}(\log\left|\frac{\partial Y}{\partial X}\right|) = (\frac{\partial Y}{\partial X})^{-1} \frac{\partial}{\partial w}(\frac{\partial Y}{\partial X}) \tag{D-4}$$

For the hyperbolic tangent function as the $g(x)$, we get:

$$\Delta w \propto \frac{1}{w} + X(1-2Y) \tag{D-5}$$

Similarly, for 2 by 2 case, we can derive the following rule:

$$\Delta \mathbf{W} \propto [\mathbf{W}^T]^{-1} - 2 \cdot \mathbf{Y} \cdot \mathbf{X} \tag{D-6}$$

Where $\mathbf{X}=\{X_1, X_2\}$ is the input vector, $\mathbf{Y}=\{Y_1, Y_2\}$ is the output vector, and $\mathbf{W}$ is 2 by 2 the matrix, so that $\mathbf{Y} = g(\mathbf{W} \cdot \mathbf{X})$.

D.2 Minimizing Kullback-Leibler (KL) divergence

Amari [101] started the derivation from the Kullback-Leibler(K-L) divergence. The cost function that was used is the K-L divergence between the joint distribution of the output ($f_{U_1,U_2}(u_1,u_2)$) and the distribution of product of the individual output

($f_{U_1}(u_1)\cdot f_{U_2}(u_2)$). If the distance is zero, it means that $f_{U_1,U_2}(u_1,u_2)=f_{U_1}(u_1)\cdot f_{U_2}(u_2)$, which is the definition of statistical independence for the elements of U_i. The K-L distance can be described as:

$$D_{f_{U_1,U_2}(u_1,u_2)\|f_{U_1}(u_1)\cdot f_{U_2}(u_2)}(\mathbf{W})=\int f_{U_1,U_2}(u_1,u_2)\log\frac{f_{U_1,U_2}(u_1,u_2)}{f_{U_1}(u_1)\cdot f_{U_2}(u_2)}du_1du_2 \tag{D-7}$$

The above equation can be substituted using the entropy definition:

$$D_{f_{U_1,U_2}(u_1,u_2)\|f_{U_1}(u_1)\cdot f_{U_2}(u_2)}(\mathbf{W})=\sum_{i=1}^{2}H(U_i)-H(\mathbf{U}) \tag{D-8}$$

Where $H(\mathbf{U})$ is the joint entropy of output $\mathbf{U}$, and $H(U_i)$ is the marginal entropy of the i-th output.

Using a Gram-Charlier cumulant expansion [119], the marginal entropy of the right hand side of the above equation can be approximated as:

$$H(U_i)\approx\frac{1}{2}\log(2\pi e)-\frac{\kappa_i(3)^2}{2\cdot 3!}-\frac{\kappa_i(4)^2}{2\cdot 4!}+\frac{5}{8}\kappa_i(3)^2\kappa_i(4)+\frac{1}{16}\kappa_i(4)^3 \tag{D-9}$$

where $\kappa_i(a)$ is the a-th order moment of the i-th output. Using Equation (D-9) and $H(\mathbf{U})=H(\mathbf{X})+\log|\det(\mathbf{W})|$, we have:

$$\begin{aligned}D_{f_{U_1,U_2}(u_1,u_2)\|f_{U_1}(u_1)\cdot f_{U_2}(u_2)}(\mathbf{W})\approx&-H(\mathbf{X})-\log|\det(\mathbf{W})|+\frac{N}{2}\log(2\pi e)\\&-\sum_{i=1}^{2}\left[\frac{\kappa_i(3)^2}{2\cdot 3!}+\frac{\kappa_i(4)^2}{2\cdot 4!}-\frac{5}{8}\kappa_i(3)^2\kappa_i(4)-\frac{1}{16}\kappa_i(4)^3\right]\end{aligned} \tag{D-10}$$

Equation (D-10) will serve as the cost function. In order to find its gradient we need perform $\frac{\partial D_{f_{U_1,U_2}(u_1,u_2)\|f_{U_1}(u_1)\cdot f_{U_2}(u_2)}(\mathbf{W})}{\partial\mathbf{W}}$, and finally we can get the learning rule as:

$$\Delta\mathbf{W}\propto[\mathbf{W}^T]^{-1}-\varphi(\mathbf{U})\cdot\mathbf{U}^T \tag{D-11}$$

Where $\varphi(x)=\frac{3}{4}x^{11}+\frac{25}{4}x^9-\frac{14}{3}x^7-\frac{47}{4}x^5+\frac{29}{4}x^3$. It is easy to see that the learning rule is almost same rule as the (D-6). The only difference is the activation function.

After obtaining the usual gradient (D-11), Amari proposed to use the natural gradient which performs the steepest descent. The natural gradient rescales the normal gradient space by right multiplying $\mathbf{W}^T\mathbf{W}$ in the both sides of equation (D-11), which gives the following:

$$\Delta\mathbf{W} \propto [\mathbf{I} - \varphi(\mathbf{U}) \cdot \mathbf{U}^T] \cdot \mathbf{W} \qquad \text{(D-12)}$$

By performing the descent using natural gradient, convergence is significantly faster and more stable. In addition to good convergence behavior, there is also increased efficiency since the learning rule does not include a matrix inversion operation.

Appendix E. Dynamic Time Warping & Uniform Time Warping

E.1 Dynamic Time Warping

The standard definition Dynamic Time Warping distance can be found [120][121].The definition of Dynamic Time Warping distance between two vectors **X** and **Y** is:

$$D_{DTW}(\mathbf{X},\mathbf{Y}) = D(X_1,Y_1) + \min\begin{cases} D_{DTW}(\mathbf{X},\mathrm{Rest}(\mathbf{Y})) \\ D_{DTW}(\mathrm{Rest}(\mathbf{X}),\mathbf{Y}) \\ D_{DTW}(\mathrm{Rest}(\mathbf{X}),\mathrm{Rest}(\mathbf{Y})) \end{cases} \quad \text{(E-1)}$$

where X_1 and Y_1 are the first element of vectors **X** and **Y**, respectively, and Rest(·) refers to rest elements without the first element.

The process of computing the DTW distance can be visualized as Figure E-1.

We construct a $n \times m$ matrix to align the vector **X** and **Y**. The ceil(i,j) corresponds to the alignment of the element X_i and Y_j. A warping path, P, from cell (1,1) to (n,m) corresponds to a particular alignment, element by element, between **X** and **Y**.

$$P = p_1 p_2, \cdots, p_L = (p_1^x p_1^y), (p_2^x p_2^y), \cdots, (p_L^x p_L^y), \quad \max(n,m) \le L \le n+m-1 \qquad \text{(E-2)}$$

The distance between **X** and **Y** on the warping path is the distance between $x_{p_t^x}$ and $y_{p_t^y}$, t=1,2,⋯,L.

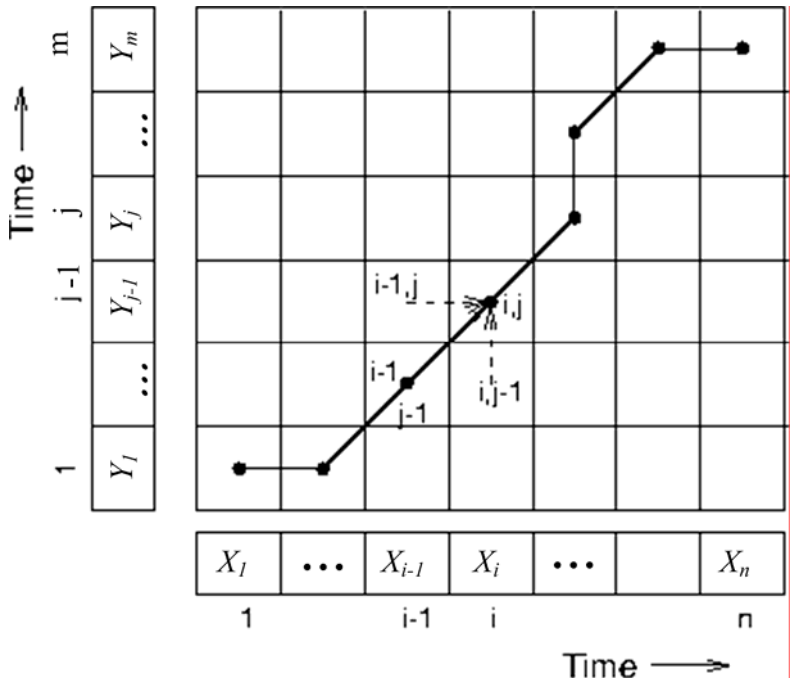


Figure E-1: Dynamic time warping for vector **X** and **Y**

The number of possible warping paths grows exponentially with the length of the vectors. The distance that is minimized over all paths is the Dynamic Time Warping distance. It can be computed using Dynamic Programming in $O(m \cdot n)$[120].

E.2 Uniform Time Warping

Uniform Time Warping(UTW) is a special case of Dynamic Time Warping. The constraint imposed by UTW is that the warping path must be diagonal.

The definition of Uniform Time Warping distance between two vectors **X** and **Y** is:

$$D_{UTW}(X,Y) = \frac{\sum_{i=1}^{mn-1} (X_{\lfloor i/m \rfloor} - Y_{\lfloor i/n \rfloor})^2}{mn} \qquad \text{(E-3)}$$

This equals to stretch both time axis of **X** and **Y** to be $m \times n$, and the comparison of two

different length vectors can be made on the normalized from. In this way, the two melodies have the different length can be compared.

Appendix F. Proportional Transportation Distance

Proportional Transportation Distance (PTD) was first proposed in [122] and has been proven efficient to measure the melodic similarity [111]. PTD is tightly related to the Earth Mover Distance (EMD).Therefore, in this appendix, we first introduce EMD, followed with PTD.

F.1 Earth Mover Distance

The Earth Mover Distance between two weighted point set measures the minimum amount of work needed to transform one into the other by moving weight. Intuitively speaking, a weighted point can be seen as an amount of earth or mass; alternatively it can be taken as an empty hole with a certain capacity. We can arbitrarily assign the role of the supplier to one set and that of the receiver/demander to the other one, setting, in that way, the direction of weight movement. The EMD then measures the minimum amount of work needed to fill the holes with earth (measured in weight

units multiplied with the covered ground distance). See Cohen's Ph.D. thesis (1999) for a more detailed description of the EMD.

Let $\mathbf{A}=\{a_1,a_2,\cdots,a_m\}$ be a weighted point set such that $a_i=\{(x_i,w_i)\}$, $i=1,2,\cdots,m$, where x_i is vertex and w_i being its corresponding weight. Let $W=\sum_{i=1}^{n} w_i$ be the total weight of set **A**. (when used in melody similarity measuring , the x_i can be considered as dual (onset time of the i-th note, its pitch) , while w_i represents the duration of the i-th note.)

The EMD can be formulated as a linear programming problem. Given two weighted point sets **A**, **B** and Eucildean distance d, we denote as f_{ij} the elementary flow of weight from x_i to y_j over the distance d_{ij}. If W,U are the total weights of **A, B** respectively, the set of all possible flows $\xi=[f_{ij}]$ is defined by the following constraints:

a) $f_{ij}\geq 0, i=1,\cdots,m, j=1,\cdots,n$

b) $\sum_{j=1}^{n} f_{ij} \leq w_i, i=1,\cdots,m$

c) $\sum_{i=1}^{m} f_{ij} \leq u_i, j=1,\cdots,n$

d) $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = \min(W,U)$

These constraints say that each particular flow is non-negative; no point from the "supplier" set emits more weight than it has, and no point from the "receiver" receives more weight than it needs. Finally, the total transported weight is the minimum of the total weights of the two sets.

The flow of weight f_{ij} over a distance d_{ij} is penalized by its product with this distance. The sum of all these individual products is the total cost for transforming **A**

into **B**. The EMD(**A**, **B**) is defined as the minimum total cost over ξ , normalized by the weight of the lighter set; a unit of cost or work corresponds to transporting one unit of weight over one unit of ground distance. That is:

$$EMD(\mathbf{A},\mathbf{B}) = \frac{\min_{F\in\xi} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}}{\min(W,U)} \qquad \text{(F-1)}$$

F.2 Proportional Transportation Distance

The EMD doesnot obey the triagular inequality[123] , which is a common property that similarity measure should have. Therefore, in [122], the author propose the PTD which is a modified EMD and is more relaible than EMD since triagular inquality still holds.

The PTD is defined as follows:

Let **A**, **B** be tow weighted point sets, W,U the total weight of **A** and **B**, and d a Eucildean distance. The set of all feasible flows $\xi = [f_{ij}]$ from **A** to **B** is defined by the following constraints:

a) $f_{ij} \geq 0, i=1,\cdots,m, j=1,\cdots,n$

b) $\sum_{j=1}^{n} f_{ij} = w_i, i = 1,\cdots,m$

c) $\sum_{i=1}^{m} f_{ij} = \frac{u_j W}{U}, j = 1,\cdots,n$

d) $\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} = W$

The PTD(**A**,**B**) is given by:

$$PTD(\mathbf{A},\mathbf{B}) = \frac{\min_{F\in\xi} \sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij} d_{ij}}{W} \qquad \text{(F-2)}$$

Constraints 2 and 4 force all of **A**'s weight to move to the positions of points in **B**.

Constraint 3 ensures that this is done in a way that preserves the old percentages of weight in **B**.

Reference

[1] A.L Uidenbogerd and J. Zobel, "Matching techniques for large music database", In *Proc. ACM International Conference on Multimedia,* 1999, pp.57-66.

[2] International Federation of the Phonographic Industry website news, http://www.ifpi.org/site-content/library/digital-music-report-2006.pdf

[3] D.J. Stephen, "Music Information Retrieval (Chapter 7)", *In Annual Review of Information Science and Technology* 37, Methord, NJ: Information Today, 2003. pp.295-340. http://music-ir.org/downie_mir_arist37.pdf

[4] B. Donald and C. Tim, "Problems of Music Information Retrieval in the Real World", In *Information Processing and Management*, 2002, Vol.38, NO.2: pp.249-272.

[5] E. Wold, T. Blum, D. Keislar and J. Wheaton, "Content-based Classification Search and Retrieval of Audio", *IEEE Multimedia*, 1996, Vol.3, NO.3, pp.27-36.

[6] G. Tzanetakis, "Manipulation, Analysis and Retrieval Systems for Audio Signal", Ph.D Thesis, *Princeton University*, 2002. http://www.cs.uvic.ca/~gtzan/papers/thesis.pdf

[7] R.Typke, F. Wiering, and R.C. Veltkamp, "A Survey of Music Information Retrieval Systems", In *Proc. International Conference on Music Information Retrieval, 2005.*

[8] J. Foote, "Content-Based Retrieval of Music and Audio", *Multimedia Storage and Archiving System II*, *Proc. SPIE*, 1997, Vol.3229: pp.138-147.

[9] E. Schierer, "Music Listening Systems", Ph.D Thesis, *Massachusetts Institute of Technology*. http://web.media.mit.edu/~eds/thesis/eds-diss-full.pdf

[10] F.R. Moore, T.D. Rossing, R. F. Moore, P.A. Wheeler, "Science of Sound, 3rd Edition", *Addison-Wesley Press*, 2001.

[11] G. Peeters, "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project", CUIDADO I.S.T. Project Report, 2004.http://www.ircam.fr/anasyn/peeters/ARTICLES/Peeters_2003_cuidadoaudiofeatures.pdf.

[12] L.R. Rabiner and B.H. Juang, “Fundamentals of Speech Recognition”, *Prentice-Hall Press*, 1993.

[13] J.J. Aucouturier and F. Pachet, “Representing Musical Genre: A State of the Art”, *Journal of New Music Research*, 2003*,* Vol.32, NO.1: pp.1-12.

[14] G. Tzanetakis and P. Cook, “Musical Genre Classification of Audio Signals”, *IEEE Transactions on Speech and Audio Processing*, 2002, Vol. 10, NO.5: pp.293-302.

[15] D. Jiang, L. Lu, H. Zhang, J. Tao and L. Cai , “Music Type Classification by Spectral Contrast Feature”, In *Proc. IEEE International Conference on Multimedia and Explore*, Lausanne, Switzerland, 2002, Vol.1:pp.113–116.

[16] E. Gomez, A. Klapuri and B. Meudic, “Melody Description and Extraction in the Context of Music Content Processing”, in *Journal of New Music Research*,, 2003, Vol.32, NO.1: pp.23-40.

[17] T. Tolonen and M. Karjalainen, “A Computationally Efficient Multi-pitch Analysis Model”, *IEEE Transactions on Speech and Audio Processing*, 2000, Vol.8, NO.6: pp.708 – 716.

[18] F. Guoyon and S. Dixon, “A Review of Automatic Rhythm Description System”, *Computer Music Journal*, 2005, Vol.29: pp.34-54.

[19] G. Tzanetakis, G. Essl and P. Cook, “Automatic Musical Genre Classification of Audio Signals”, In *Proc. International Symposium on Music Information Retrieval*, Bloomington, Indiana, USA, 2001, pp.205-210.

[20] G. Tzanetakis, G. Essl and P. Cook, “Audio Analysis using the Discrete Wavelet Transform”, In *Proc. Conference in Acoustics and Music Theory Applications*, WSES, 2001.

[21] T. Li, M.Ogihara and Q.Li, “A Comparative Study on Content-Based Music Genre Classification”, In *Proc. ACM Conference on Research and Development in Information Retrieval,* Toronto, Canada, 2003, pp.282-289.

[22] D. Pye, “Content-Based Methods for the Management of Digital Music”, In *Proc. IEEE International Conference on Audio, Speech and Signal Processing,* Istanbul, Turkey, 2000, Vol.4: pp.2437-2440.

[23] M. Grimalidi, A. Kokaram and P. Cunningham, “Classifying Music by Genre Using a Discrete Wavelet Transform and a Round-Robin Ensemble”, Work report. Trinity College, University of Dublin, Ireland, 2003.

[24] E. Pampalk, A. Flexer and G.Widmer, “Improvements of Audio-Based Music Similarity and Geenre Classification”, In *Proc. International Symposium on Music Information Retrieval*, London, UK, 2005.

[25] F. Pachet and D. Cazaly, "A Taxonomy of Musical Genre", In *Proc. Content-Based Multimedia Information Access Conference*, Paris, France, 2000.

[26] F. Pachet, G. Weatermann and D. Laigre, "Musical Data mining for Electronic Music Distribution", In *Proc. Wedel Music Conference*, Italy, 2001.

[27] S. Lippens, J.P. Martens, M. Leman, B. Baets, H. Mey and G. Tzanetakis, "A Comparison of Human and Automatic Musical Genre Classification", In *Proc. IEEE International Conference on Audio, Speech and Signal Processing,* 2004, pp.233-236.

[28] J. Foot and S. Uchihashi, "The Beat Spectrum: A New Approach to Rhythm Analysis", In *Proc. IEEE International Conference on Multimedia and Explore*, Tokyo, Japan, 2001, pp.881-884.

[29] T. Joachims, "Text Categorization with Support Vector Machines", In *Proc. European Conference on Machine Learning*, *Springer-Verlag*, 1998.

[30] C. Papageorgiou, M. Oren and T. Poggio, "A General Framework for Object Detection", In *Proc. International Conference on Computer Vision*, Bombay, India, 1998, pp.555-562.

[31] T. Li and M. Ogihara, "Music Genre Classification with Taxonomy", In *Proc. IEEE International Conference on Audio, Speech and Signal Processing,* Philadelphia, PA, USA, 2005, Vol.5: pp.197-200.

[32] C.N. Maddage, C. Xu., M.S. Kankanhalli and X. Shao, "Content-Based Music Structure Analysis with the Applications to Music Semantic Understanding", *In ACM Multimedia Conference 04*, New York, 2004, pp.112-119.

[33] S. Young, etc. The HTK Book (for HTK Version 3.2).http://htk.eng.cam.edu/, *Engineering Department, Cambridge University*, December 2002.

[34] J. Bilmes, "A Gentle Tutorial on the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models", Technical Report, *University of Berkeley*, ICSI-TR-97-021, 1997. http://crow.ee.washington.edu/people/bulyko/papers/em.pdf

[35] R.O. Duda, P.E. Hart, and D.G. Stork, "Pattern Classification (Second Edition)", *A Wiley-Inter Science Publication*, 2000.

[36] I. Mani and M.T. Maybury, "Advances in Automatic Text Summarization", Combridge, Massachusetts: *MIT Press*, 1999.

[37] C. Hori and S. Furui, "Improvements in Automatic Speech Summarization and Evaluation Methods", In *Proc. International Conference on Spoken Language Processing* ,Beijing, China, 2000, Vol. 4: pp.326-329.

[38] R. Kraft, Q. Lu and S. Teng, "Method and Apparatus for Music Summarization and Creation of Audio Summaries", *US Patent* 6,225,546

[39] B. Logan and S. Chu, "Music Summarization Using Key Phrases", In *Proc. IEEE International Conference on Audio, Speech and Signal Processing,* Istanbul, Turkey, 2000, Vol.2: pp.II749 - II752.

[40] C. Xu, Y. Zhu, and Q. Tian, "Automatic Music Summarization Based on Temporal, Spectral and Cepstral Features", In *Proc. of IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, 2002, Vol.1: pp.117–120.

[41] L. Lu and H. Zhang, "Automated Extraction of Music Snippets", In *Proc. ACM International Conference on Multimedia*, Berkeley, CA, USA, 2003, pp.140-147.

[42] M. Cooper and J. Foote, "Automatic Music Summarization via Similarity Analysis", In *Proc. International Conference on Music Information Retrieval*, Paris, France, 2002, pp.81-85.

[43] J. Foote, M. Cooper and A. Girgensohn, "Creating Music Video using Automatic Media Analysis", In *Proc. ACM international conference on Multimedia*, Juan-les-Pins, France, 2002, pp.553-560.

[44] M.A. Bartsch and G.H. Wakefield, "To Catch a Chorus: Using Chroma-based Representations for Audio Thumbnailing", *In Proc. Workshop on Applications of Signal Processing to Audio and Acoustics*(WASPAA), New Paltz, New York, 2001,pp.15–18.

[45] W. Chai and B. Vercoe, "Music Thumbnailing via Structural Analysis", In *Proc. ACM international conference on Multimedia*, Berkeley, CA, USA, 2003, pp.223-226.

[46] D. Yow, B.L. Yeo, M. Yeung and G. Liu, "Analysis and Presentation of Soccer Highlights from Digital Video", In *Proc. Asian Conference on Computer Vision*, Singapore, 1995, Vol. II: pp.499-503.

[47] D. Tjondronegoro, Y.P. Chen and B. Pham, "Sports Video Summarization Using Highlights and Play-Breaks", *Proc. the 5th ACM SIGMM international workshop on Multimedia information retrieval,* Berkeley, California, US, 2003, pp.201-208.

[48] Y. Nakamura and T. Kanade, "Semantic Analysis for Video Contents Extraction–Spotting by Association in News Video", In *Proc. ACM International Multimedia Conference*, Seattle, Washington, USA, 1997, pp.393-401.

[49] Y. Gong, X. Liu and W. Hua, "Creating Motion Video Summaries with Partial Audio-Visual Alignment", In *Proc. IEEE International Conference on Multimedia and Expo*, Lausanne, Switzerland, 2002, Vol.1: pp.285–288.

[50] Y. Gong, X. Liu and W. Hua, “Summarizing Video by Minimizing Visual Content Redundancies”, In *Proc. IEEE International Conference on Multimedia and Expo*, Tokyo, Japan, 2001, pp.788-791.

[51] J. Foote, M. Cooper and A. Girgensohn, “Creating Music Videos Using Automatic Media Analysis”, In *Proc. ACM international conference on Multimedia*, Juan-les-Pins, France, 2002, pp.553-560.

[52] S. Pfeiffer, R. Lienhart, S. Fischer and W. Effelsberg, “Abstracting Digital Movies Automatically”, *Journal of Visual Communication and Image Representation*, 1996, Vol.7, NO.4: pp.345-353.

[53] L. Agnihotri, N. Dimitrova, J. Kender and J. Zimmerman, ”Music Videos Miner”, In *Proc. ACM International Conference on Multimedia,* Berkeley, CA, USA, 2003, pp.442-442.

[54] L. Agnihotri, N. Dimitrova, J. Kender, “Design and Evaluation of a Music Video Summarization System”, In *Proc. IEEE International Conference on Multimedia and Expo*, Taibei, Taiwan, 2004, pp. 1943-1946.

[55] S. Gao, N.C. Maddage and C.H. Lee, “A Hidden Markov Model Based Approach to Music Segmentation and Identification”, In *Proc. IEEE Pacific-Rim Conference on Multimedia*, Singapore, 2003, pp.1576–1580.

[56] J.R. Deller, J.H.L. Hansen and J.G. Proakis, “Discrete-Time Processing of Speech Signals”, *Wiley-IEEE Press*, September 1999.

[57] T. Zhang, “Automatic Singer Identification,” In *Proc. IEEE Conference on Multimedia and Expo*, Baltimore, Maryland, USA, 2003, pp.33-36.

[58] N.C. Maddage, C. Xu and Y. Wang, “A SVM–based Classification Approach to Musical Audio”, In *Proc. of International Conference on Music Information Retrieval*, Baltimore, Maryland, USA, 2003, pp.243-244.

[59] N. Eugene, “The Analysis and Cognition of Basic Melodic Structures”, *University of Chicago Press*, 1990.

[60] E.D. Scheirer, “Tempo and Beat Analysis of Acoustic Musical Signals”, *Journal of the Acoustical Society of America*, 1998, Vol.103, NO.1: pp.588-601.

[61] X Sun, A. Divakaran and B.S. Manjunath, “A Motion Activity Descriptor and its Extraction in Compressed Domain”, In *Proc. IEEE Pacific-Rim Conference on Multimedia*, Beijing, China, 2001, pp.450-457.

[62] H. Zettl, “Sight Sound Motion: Applied Media Aesthetics (Third Edition)”, *Wadsworth publishing company*, 1999.

[63] Y. Sugana , and S. Iwamiya, “The Effects of Audio-Visual Synchronization on the Attention to the Audio-Visual Materials”, Multimedia Modeling, Shuji Hashimoto ed., *World Scientific*, 2000, pp.1-17.

[64] T.H. Cormen, C.E. Leiserson, R.L. Rivest and C. Stein, "Introduction to Algorithms (Second Edition)", *MIT Press* 4th Printing, 2001.

[65] C. Yang, "Efficient Acoustic Index for Music Retrieval with Various Degrees of Similarity", In *Proc. of the tenth ACM international conference on Multimedia*, Juan-les-Pins, France,2002, pp. 584 – 591.

[66] W. Birmingham, C. Meek, K. OMalley, B. Pardo and J. Shifrin, "Music Information Retrieval Systems", *Dr. Dobb's Journal*,2003.

[67] A. Ghias, J. Logan, D. Chamberlin and B.C. Smith, "Query by Humming", In *Proc. ACM Multimedia 95*, San Francisco, USA, 1995, pp.231-236.

[68] R. MaNab, L. Smith, I. Witten, C. Henderson and S. Cunningham, "Towards Digital Music Library: Tune Retrieval from Acoustic Input", In *Proc. Digital Library'96*, 1996, pp.11-18.

[69] A. Chen, M. Chang, J. Chen, J.L. Hsu and S. Hua, "Query by music segments: an efficient approach for song retrieval", In *Proc. IEEE International Conference on Multimedia and Explore*, New York City, NY, USA, 2000, pp.889-892.

[70] Y. Zhu, M. S. Kankanhalli and C. Xu, "Pitch Tracking and Melody Slope Matching for Song Retrieval", In *Proc. IEEE Pacific-Rim Conference on Multimedia*, Beijing, China, 2000, pp.530-537.

[71] A.L.P. Chen, M. Chang and J .Chen, "Query by Music Segments: An Efficient Approach for Song Retrieval", In *Proc. IEEE International Conference on Multimedia and Explore*, 2000, pp.873-876.

[72] C. Francu and C.G. Nevill-Manning, "Distance Metrics and Indexing Strategies for Digital Library of Popular Music", In *Proc.* IEEE *International Conference on Multimedia and Explore*, 2000, pp.889-892.

[73] J. Rouat, Y. C. Liu, and D. Morissette, "A pitch determination and voiced/unvoiced decision algorithm for noisy speech" *Speech Communication*, 1997, Vol. 21: pp.257-260.

[74] K. Dressler, "Extraction of the Melody Pitch Contour from Polyphonic Audio", MIREX 2005 Contest, MIR 2005. http://www.music-ir.org/evaluation/mirex-results/articles/melody/dressler.pdf

[75] H.Malik, A. Khokhar, R.Ansari and B.C. Baillon, "Predominant Pitch Contour Extraction from Audio Signals", In *Proc.* IEEE *International Conference on Multimedia and Explore*, 2002,

[76] M.Goto, "A predominant- F0 estimation method for real- time detection of melody and bass lines in CD recordings", *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2000, pp.II757-II760.

[77] J. Song, S.Y. Bae and K. Yoon, "Mid-Level Melody Representation of Polyphonic Audio for Query-by-Humming System", *Proc. International Symposium on Music Information Retrieval*, 2002.

[78] A.J. Bell and T.J. Sejnowski. "An Information-Maximization Approach to Blind Separation and Blind Deconvolution", *Neural Computation*, 1995, Vol.7: pp.1129-1159.

[79] K. Torkkola. "Blind Separation of Convolved Sources Based on Information Maximization", In *Proc. IEEE Workshop on Neural Networks for Signal Processing*, 1996, pp.423–432.

[80] T. W. Lee and A.J. Bell "Blind Separation of Delayed and Convolved Sources", *Advances in Neural Information Processing Systems*, 1996, Vol.9: pp. 758-764.

[81] P. Smaragdis, "Information Theoretic Approaches to Source Separation", M. Sc. Thesis, *MIT Media Lab*, June 1997. http://sound.media.mit.edu/~paris/paris-msc.ps.gz

[82] L. Parra and C. Spence, "Convolutive Blind Source Separation of Non-Stationary Sources", *IEEE Trans. Speech Audio Processing*, 2000, Vol.8, NO.3: pp.320-327.

[83] N. Mitianoudis and M.E. Davies, "Audio Source Separation of Convolutive Mixtures", *IEEE Trans. Speech Audio Processing,* 2003, Vol.11, NO.5: pp.489-497.

[84] A. Dapena and C. Serviere, "A Simplified Frequency-Domain Approach for Blind Separation of Convolutive Mixtures", *Proc. ICA* 2001, San Diego, USA, 2001, pp. 569--574.

[85] S. Haykin and Z. Chen, "The Cocktail Party Problem", *Neural Computation.* 2005, Vol.17: pp.1875-1902.

[86] J.L. Lacoume, "A Survey of Source Separation", *In Proc. International Conference on Independent Component Analysis* (ICA), Aussois, France, 1999, pp.1-6.

[87] A. Hyvarinen, "Survey on Independent Component Analysis", *Neural Computing Surveys*, 1999, Vol.2: pp.94--128.

[88] T.-W. Lee, "Independent Component Analysis: Theory and Applications", *Kluwer Academic Publishers*, 1998.

[89] K.H. Knuth, "A Bayesian Approach to Source Separation". In *Proc. of the First International Workshop on Independent Component Analysis and Signal Separation*, Aussios, France, 1999, pp.283-288.

[90] J.F. Cardoso, "Informax and Maximum Likelihood for source Separation", *IEEE Letters on Signal Processing*, 1997, Vol.4: pp.112-114.

[91] P.J. Walmsley, "Signal Separation of Musical Instruments", Ph.D. Thesis, *University of Cambridge*, U.K., 2000. http://www-sigproc.eng.cam.ac.uk/oldhomes/pjw42/public_html/ftp/fyrep1.pdf

[92] L. Molgedey and H. Schuster, "Separation of Independent Signals Using Time-Delayed Correlations", *Physical Review Letters*, 1994, Vol.72, NO.23: pp.3634-3637.

[93] F. Ehler and H. Schuster, "Blind Separation of Convolutive Mixtures and an Application in Automatic Speech Recognition in Noisy Environment", In *IEEE Transactions on Signal processing*, 1997, Vol.45, NO.10: pp.2608-2609.

[94] T.W. Lee and A. Ziehe, "Combining Time-Delayed Decorrelation and ICA: Toward Solving the Cocktail Party Problem", In *Proc. IEEE International Conference on Audio, Speech and Signal Processing,* Seattle, WA, May 1998, pp.1089-1092

[95] R. Lambert, "Multi-Channel Blind Deconvolution: FIR Matrix Algebra and Separation of Multi-Path Mixtures". Ph.D Thesis, *University of Southern California, Department of Electrical Engineering*. http://www.dcs.shef.ac.uk/~ljupco/papers/lambert-mydis.ps.gz

[96] A. Belouchrani and M.G. Amin, "Blind Source Separation Based on Time-Frequency Signal Representations", *IEEE Transactions on Signal Processing*, November, 1998, Vol.46, NO.11: pp.2888-2897.

[97] P. Bofill and M.Zibulevsky, "Blind Separation of More Sources than Mixtures using Sparsity of Their Short Time Fourier Transform", In *Proc. International Workshop on Independent Component Analysis and Signal Separation*, Helsinki, Finland, 2000, pp.87-92.

[98] P.Bofill, "Underdetermined Blind Separation of delayed Sound Sources in the Frequency Domain", *Neural Computation*, 2003, Vol.55, NO.3/4: pp.627-641.

[99] Y. Özgür and R. Scott, "Blind Separation of Speech Mixtures via Time-Frequency Masking", *IEEE Transactions on Signal Processing*, July, 2004, Vol.52, NO.7: pp.1830-1847.

[100] I.T. Jolloffe, "Principle Component Analysis", *Springer-Verlag*, 1986.

[101] S. Amari, A. Cichocki, and H.H. Yang, "A New Learning Algorithm for Blind Source Separation", In *Advances in Neural Information Processing Systems*, Cambridge, MA., MIT Press, 1996, pp.757-763.

[102] J.M. Eagle, "Handbook of Recording Engineering, Fourth Edith", *Kluwer Academic Publisher*, 2002.

[103] K. Torkkola. "Blind Separation of Delayed Sources Based on Information Maximization", In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Atlanta, GA, USA, 1996, pp.3509–3512.

[104] T.W. Lee, A.J. Bell and R. Orglmeister, "Blind Source Separation of Real World Signals", In *Proc. International Conference of Neural Networks*, 1997.

[105] N. Murata and S. Ikeda, "An On-line algorithm for Blind Source Separation on Speech Signals", *Proc. International Symposium on Nonlinear Theory and its Applications*, 1998.

[106] A. Dapena and C. Serviere, "A Simplified Frequency domain Approach for Blind Separation of Convolutive Mixtures", In *Proc. International Workshop on Independent Component Analysis and Signal Separation*, San Diego, California, USA., 2001, pp.569-574.

[107] M,Ikram and N.Murata, "Exploring permutation inconsistency in blind separation of speech signals in a reverberant environment", In *IEEE International Conference on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, 2002, Vol.2: pp.1041-1044.

[108] C. Duxburg, M. Sandler., and M. Davies. "A Hybrid Approach to Musical Note Onset Detection", In *Proc. International Conference on DAFx.* 2002.

[109] Y. Zhu, M.S. Kankanhalli and Q. Tian, "Similarity Matching of Continuous Melody Contours for Humming Query of Melody Databases", In *IEEE Workshop on Multimedia Signal Processing 02*, 2002, pp.249-252.

[110] Y. Zhu and D. Shasha "Query by Humming: a Time Series Database Approach", In *Proc*. *ACM SIGMOD 2003,* 2003, pp.181-192.

[111] R. Typke, P. Giannopoulos, R.C. Veltkamp, F. Wiering and R. OOstrum, "Using Transportation Distances for Measuring Melodic Similarity", In *Proc. Int. Sym. on Music Information Retrieval03*, 2003*,* pp.107-114.

[112] J.P. Chin, V.A. Diehl, and K.L. Norman, "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface", *Proc. of SIGCHI Conference on Human Factors in Computing System.*, Washington, D.C., USA,1988, pp.213-218.

[113]G. Jang and T. Lee, "A Maximum Likelihood Approach to Single-channel Source Separation", *Journal of Machine Learning Research*, 2003 Vol.4, NO.1: pp.1365-1392.

[114] T. Leung and C. Ngo, "Indexing and Matching of Polyphonic Songs for Query-by-Singing System", In *Proc. ACM international conference on Multimedia 04,* New York*,* NY, USA, 2004, pp.308-311.

[115] T. Yoshizawa, H. Schweitzer, "Long-Term Learning of Semantic Grouping from Relevance-Feedback", In *Proc*. *ACM Multimedia workshop on multimedia databases*, New York, NY, USA, 2004, pp.165-172.

[116] Tat-Seng Chua, Chunxin Chu and Mohan S. Kankanhali. "Relevance Feedback Techniques for Image Retrieval Using Multiple Attributes." *Proc. of IEEE International Conference on Multimedia Computing and Systems* (ICMCS'99). Florence, Italy. Jun 1999, pp.890-894.

[117] S. Haykin, "Neural Networks, A Comprehensive Foundation, 2nd Edition", *Prentice Hall Press*, 1999.

[118] A. Papoulis, "Probability, Random Variables and Stochastic Processes, Second Edition", *McGraw-Hill* publishing, New York, 1984.

[119] A. Stuart and K. Ord, "Kendall's Advanced Theory of Statistical, Vol.1 Six Edition", *Halsted Press*, New York, 1994.

[120] D. Berndt and J. Clifford, "Using Dynamic Time Warping to Find Patterns in Time Series", In *Advances in Knowledge Discovery and Data Mining*, AAAI ,MIT press,1994, pp.229-248.

[121] B.K. Yi, H.V. Jagadish and C. Faloutsos, "Efficient Retrieval of Similar Time Sequences under Time Warp", In Proc. *International Conference on Data Engineering*,1998, pp.201-208.

[122] P. Giannopoulos and R.C. Veltkamp, "A Pseudo-metric for Weighted Point Sets", In Proc. *7th European Conf. Comp. Vision*, *LNCS 2352*, 2002, pp. 715-731.

[123] S. Cohen. "Finding Color and Shape Patterns in Images", Ph.D. Thesis, *Stanford University, Department of Computer Science*, 1999. http://vision.stanford.edu/public/publication/cohen/cohenTr99.ps.gz

Publications

- **Journals**

1. C. Xu, N. C. Maddage and X. Shao, "Automatic Music Classification and Summarization", *IEEE Trans on Speech & Audio*, Vol.13, NO.3, May, 2005, pp.441-450.
2. X. Shao， C. Xu, N. C. Maddage, M. S. Kankanhalli, Q. Tian and J.S. Jin, "Automatic Summarization of Music Videos", *ACM Transactions on Multimedia Computing, Communications, and Applications* (TOMCCAP), Vol.2, NO.2, May, 2006,pp.1-22.

- **Book Chapters**

1. C. Xu, X. Shao, N. C. Maddage, J. S. Jin, Q. Tian, "Content-Based Music Summarization and Classification", In *Managing Multimedia Semantics* ,Independent Pub Group, Feb , 2005.

- **Conference Papers**

1. X. Shao, C. Xu, M. S. Kankanhalli, "Automatically Generating Summaries for Musical Video", in *International Conference of Image Processing* (ICIP03),Barcelona,Spain,2003.
2. C. Xu, N. C. Maddage, X. Shao, Q. Tian, "Musical Genre Classification Using Support Vector Machines", in *Proc. International Conference of Acoustics ,Speech& Signal Processing* (ICASSP03), Hong Kong, China,2 003.
3. X. Shao, C. Xu, M. S. Kankanhalli, "Applying Neural Network on Content Based Audio Classification", in *Proc. IEEE Pacific-Rim Conference on Multimedia* (PCM03), Singapore, 2003.
4. X. Shao, C. Xu, Y. Wang, M. S Kankanhalli, "Automatic Music Summarization in Compressed Domain", in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP04), Montreal, Canada, 2004.
5. C. Xu, X. Shao, N. C. Maddage, M. S. Kankanhalli, Q. Tian, "Automatically Summarize Musical Audio Using Adaptive Clustering", in *Proc. IEEE*

International Conference of Multimedia Explore (ICME04), Taipei, Taiwan, China, 2004.

6. X. Shao, C. Xu, M. S. Kankanhalli, "Unsupervised Classification of Music Genre Using Hidden Markov Model", in *Proc. IEEE International Conference of Multimedia Explore* (ICME04), Taipei, Taiwan, China, 2004.
7. X. Shao, C. Xu, M. S. Kankanhalli, "A New Approach to Automatic Music Video Summarization", In *IEEE International Conference of Image Processing* (ICIP04), Singapore, 2004.
8. N. C. Maddage, C. Xu, M. S. Kankanhalli, X. Shao, "Content-based Music Structure Analysis with the Applications to Music Semantic Understanding", in *Proc. ACM Multimedia Conference*(ACM MM04).
9. X. Shao, N. C. Maddage, C. Xu, M. S. Kankanhalli, "Automatic Music Summarization Based on Music Structure Analysis", in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing* (ICASSP05), Philadelphia,USA,2005.
10. C. Xu, X. Shao, N. C. Maddage, M. S Kankanhalli, "Automatic Music Video Summarization Based on Audio-Visual-Text Analysis and Alignment", in *Proc.* 28th *Annual ACM SIGIR*, Salvador, Brazil,2005
11. X. Shao, C. Xu, M. S Kankanhalli, "Predominant Vocal Pitch Detection in Polyphonic Music", in *Proc.* IEEE *International Conference of Multimedia Explore* (*ICME06*) ,Toronto, Ontario,Canada,2006.