MUSIC RETRIEVAL SYSTEMS: ROBUST PERFORMANCE UNDER THE EFFECT OF UNCERTAINTY

by

Erdem Unal

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(ELECTRICAL ENGINEERING)

August 2008

Dedication

to my family

Acknowledgments

I would like to acknowledge the support of the the National Science Foundation (NSF) which made this research possible.

I would also like to express my gratitude to my advisors, Shrikanth Narayanan, Elaine Chew and Panayiotis Georgiou whose advice and understanding made this work possible.

Finally, I would like to acknowledge numerous constructive discussions I have had with my colleagues Maverick Shih, Samuel Kim and Abhinav Sethy among others - and would also like to thank them for making my experience in graduate school that much more enjoyable.

Table of Contents

List of Figures

List of Tables

Abstract

Music Information Retrieval (MIR) is gaining widespread attention and becoming increasingly important. The growing capacity of web servers parallels the explosion of information generated worldwide. The need for efficient and natural access to these databases cannot be overstated. Digital music and its associated information are prime examples of such complex information that can be stored in a variety of formats, such as MP3, MIDI, wav, scores etc. These data can also be accessed in multiple ways. If the user is familiar with the name of the song or the band, and the source material is annotated with metadata, retrieval can be straightforward. However, if one does not know the lyrics, title, or the performer, alternative retrieval methods are necessary, such as through singing, humming, or playing a sample of the piece as a the query to the database. Enabling such kinds of natural human interactions with large databases has thus become an essential component of effective and flexible MIR systems.

In this thesis, two general domains for MIR systems are under discussion: a) Retrieval in Monophonic Music, and b) Retrieval in Polyphonic Music. For both domains, this thesis investigates the different sources and effects of uncertainty that is present in the input level, and system level, and present algorithms for solving the robust retrieval problem by combining music knowledge, signal processing techniques and statistical analysis.

First, we discuss Query by Humming, a specific instance of music retrieval systems in monophonic music domain, where only one sound source is available at a time. Here, straight forward signal analysis via prosodic features such as pitch and energy can be used to achieve accurate transcription from audio to symbol domain, however, the variability in the way people produce humming is not easy to handle with such straight forward algorithms. Since the transcription provided by the system front end will be used in the query engine, the robustness against user dependent variability is important. The performance of the transcription directly affects the performance that of the retrieval engine.

Our approach for achieving robust performance under the effect of uncertatinty is statistical. We first discuss our experiments for collection of real world humming data. The data is important in designing statistical systems. The goal is to achieve a collection of data that represents the general variability that is expected in the input of QBH systems. The data is also used in estimating important parameters of the front end of the system for segmentation, and also it is used in testing our QBH system's retrieval performance. We analyzed the humming performance of different users against different criteria such as, the effects of musical background, musical structure of the target melody, and familiarity. We also tried to observe performance differences over humming different interval levels such as high intervals vs low intervals, perfect intervals vs augmented intervals etc... The final goal is to use the acquired statistical information as a guidence through our retrieval calculations.

An Hidden Markov Models (HMM) based speech recognition system is used in the front end of our QBH system. The goal is to segment humming syllables that represent musical notes in the input audio. Accurate segmentation leads accurate representation of the audio in the symbol domain. We use relative information of change in pitch and duration for consecutive notes to ensure key and rhythm independent representation.

From the two dimensional transcription of pitch contour and duration ratios, we extract fixed length characteristic finger prints(FP) from the audio at rare pitch movements and duration movements where highest and lowest change in the input is performed. The main assumption we use is a subsequence of pitch contour and duration ratios would be enough for representing a melody. These subsequences, finger prints are mapped onto the database entries and compared to see if any similarities can be found. Statistical measures are used to define and calculate the similarity distance from the extracted finger prints and the database entries to achieve robust performance.

We also extended the MIR problem to a next level, which is retrieval in polyphonic domain. Now the retreival task is performing matching between audio files, that has unlimited sound sources, and they might be in different forms with respect to expressive parameters and orchestration. In polyphonic music, since the number of instruments playing at a time and their identity is unknown, mapping the audio signal into a true note transcription is a hard task to achieve. Researchers used different machine learning techniques and they were only able to report around 55% note detection accuracy in monotimbral domain. Here a mid level representation whose performance is not affected by different spectral characteristics of the different instruments should be defined. We used a representation technique that maps small audio frames into a symbolic representation that tracks general tonal movement, behaviour and characteristics of the polyphonic audio. The selected representation is a string sequence of lexical chords for each major and minor chord of twelve distinct sounds in a full octave. The representation is achieved by mapping the audio spectrum of the individual frames onto the Spiral Array, a 3d space for tonality, that has specific tonal marks at specific coordinates. The Spiral Array is updated with respect to the tonal labeling task for faster transcription. From the mapping, a decision is made for identifying which tonal cluster the audio frame belongs to, and appropriate labeling is perfromed. The transcription process is continuously labeling consequtive audio frames with the most appropriate (closest to the tonal center).

For modeling, we used sequential statistical models, which are n-grams. An n-gram is a sub-sequence of n items from a given sequence. The n-gram sequential modeling strategy can be applied to the tonal sequences that are transcribed from polyphonic audio for statistically representing tonal movements. This sequential representation technique is similiar to the ones that is used in genetic analysis, instead of protein names, we have chord names in our sequential code. N-grams are extracted from tonal string sequences to create a statistical model for each of the polyphonic melody that is in our melody database. After appropriate smoothing, which is needed to compansate for different audio lengths, the smoothed n-grams will ne accumulated in the melody database. For retrieval, a symbolic sequence is compared to each of the smoothed n-grams in the database using perplexity based scoring. Perplexity calculates the cross enthrophy between a query sequence and the smoothed n-grams in the database. The sequential models which are close to the query sequence will be less surprised from the subsequences generated by the query, so their perplexity score will be less, which will be used as the similarity metric in our retrieval calculations.

Erdem Unal

Los Angeles, California

May 2008.

Chapter 1

Introduction

1.1 Music Information Retrieval: analysis, indexing and retrieval

Due to advances in computer and network technologies, multimedia analysis applications gained great attention by researchers. With the growing capacities of the web-servers and hand-held devices, now it is possible to store large amounts of multimedia data, for commercial and personal usage. Music data is one of the most popular forms of this data that can be stored and organized in these storage devices. There is always a need to access a certain music file with great efficiency and accuracy. This thesis investigates different ways of accessing the required data using signal level information where interactions between humans users and these databases are enabled.

Music Information Retrieval (MIR) research focuses on content based analysis indexing and retrieval of such data. Music analysis and indexing type of research mostly focuses on extracting musical features from available data for content understanding and can be categorized apart from retrieval. Beat tracking, rhythm extraction, chord and key extraction, instrument tracking, automatic accompaniment are a few example of music content analysis research topics. On the other hand, retrieval research mostly focuses on audio similarity assessment for efficient matching. Most of the time, the stored data is tagged with metadata and in the cases, that the human user has information about this metadata, retrieval can be straightforward by using text based search engines for efficient matching. On the other hand, for the cases where the human user does not have any particular information about the tagged data, the ways of retrieval becomes limited to interactions that are enabled by the retrieval systems. In this thesis, the two main problem under discussion are performing retrieval by using a real sample of the target music piece, (query by example) and by using a human produced audio sample (query by humming).

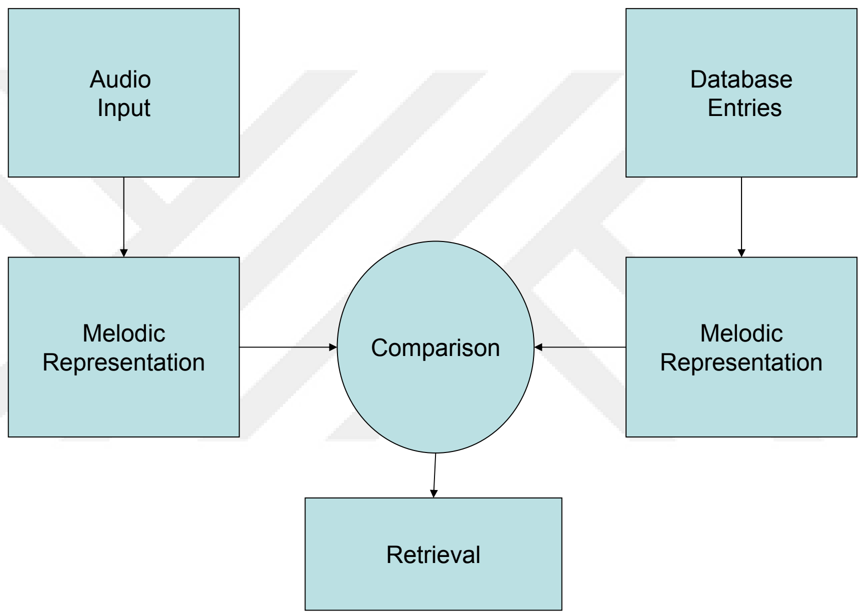


Figure 1.1: A general diagram for MIR Systems: Comparison of melodic representations of auido input and database entries

Figure 1.1 shows the big picture for a general music retrieval application, where a mid level representation is chosen to account for the content of the music, and this representation is applied to both the input and the entries in the database, so that an efficient comparison can be made in the matching and the retrieval phase of the system.

In this proposal, two cases of music retrieval applications will be examined:

- Retrieval of Monophonic Music: Query by Humming

- Retrieval of Polyphonic Music: Query by Example

One of the main focuses of this thesis is to handle different effects of variability that is present within the input caused by the production of the audio input, and moreover, caused by the system level errors that is generated while creating the melodic representation of the audio input. This variability is called the uncertainty and in this proposal, the effects of uncertainty in different forms will be analyzed.

In query by humming, the domain is limited to monophonic input, since in human humming; only one source of sound is available. As seen from the figure, the first step on constructing an MIR system is to choose an appropriate representation schema both for the input and for the database elements. Here, since the sound source is single, appropriate signal processing techniques could manage to extract required prosodic features such as pitch and energy and map the spectrum into a symbolic musical representation, mostly the musical notes or note transitions. This same representation is also accessible in the database, and finally an appropriate comparison technique needs to be employed in order to find a match between the monophonic input and the database elements that accounts for the variability that is present within the input.

For query by example, the domain is polyphonic music and it is more complex, since the number of sources in the audio is not known, and this complexity becomes hard to deal while efficiently representing the audio input. For polyphonic audio, mapping from the spectrum to an appropriate representation is also required, however, expected to be a complex task considering the multi dimensionality of the sound sources available in the audio. Usually a mid level representation is more appropriate in this case of complex scenarios. For retrieval, the selected representations of the input and the database entries are also compared and an appropriate match is selected as the result. Also, the various variations of a single music piece should be considered as the uncertainty in the audio input, similar to the variability/error in the query by humming case.

1.2 Contributions

We developed 2 separate systems as a solution for monophonic and polyphonic retrieval problems. For monophonic retrieval, we present a query by humming system that takes a user's humming as the input query and searches the monophonic audio to match with similar melodies in the database. The uncertainty in the input is the variability of the humming query, due to different humming production skills of the users. The specific contributions to this area are as follows.

- A large scale database of real world humming samples.
- HMM based front-end for accurate audio to symbol transcription
- A statistical retrieval engine

For retrieval of polyphonic music, a separate system is presented in this proposal that accepts a polyphonic audio as an input. Since a music piece can be played by different artists with different expressive variations, the variability, the uncertainty expected in the audio is more structered. The goal of the retrieval system is to find out similiar matches from a database, especially to match with expressive variations of the query audio.

- Applying a tonal space in 3D (Spiral Array and its updated form) for tracking and labeling the harmonic movement of polyphonic audio
- A mid level representation technique for polyphonic music
- Sequential modeling and retrieval of polyphonic music

This proposal presents detailed information on each of the components of the 2 systems designed for retrieval of monophonic music, and polyphonic music.

1.2.1 A large scale database of real world humming samples

In designing our query by humming system, a specific instance of music retrieval in monophonic domain, our approach is statistical. For statistical algorithms to function properly, data is crucial for training models and deducting important information about the variability of the input. In order to create a collection for representing the variability that is expected in the humming input, a large scale humming data collection experiment was performed. A sample list of melodies with respect to their musical structure is selected and participants from different musical backgrounds were asked to hum a number of melodies which they were most familiar with. In total 100 people participated in our experiments and on average 25 humming samples were recorded from each other, resulting in around 2500 recordings. The collected data is then analyzed for deducting statistical information that guides various sub-systems' for adopting their functionality against expected variability in the data.

1.2.2 HMM based front-end for accurate audio to symbol transcription

As seen from the Figure 1.1, the first task in designing music retrieval systems is to represent the audio input and the database entries in an appropriate representation so that, during retrieval, a meaningful comparison between the input and the database elements can be performed. In query by humming, since domain is monophonic, a proper representation should take into account the well structured form of the audio spectrum, and this can be used advantageously. In this work, humming is treated as being a form of speech. To capture and segment the syllables dedicated for musical notes in the humming, an HMM based speech recognizer is designed. For this, a proper selection of dictionary is needed and according to the supplied dictionary, system detects and segments the humming notes from each other. Applying a set of post processing steps using energy and pitch features, the segmentation is finalized. From the segmented audio chunks, system extracts pitch and energy information that is required for note labeling using musical symbols. The symbolic representation that is used gives relative information about the tonal and rhythmic change of the consecutive notes.

1.2.3 A statistical Retrieval Engine

Using the selected symbolic transcription, the final goal is to perform meaningful comparison of the audio input against the database elements. In this proposal, we present a finger print (FP) algorithm that extracts salient melodic information from the input that represents characteristic changes in the melodic contour with respect to pitch and duration. Selected finger prints cover fixed length melodic sequences of these characteristic changes that rarely occur in the melody however, carries more representative information about the hummed melody. The selected melodic windows, namely the finger prints, are then aligned on the database melodies and a similarity distance measure is calculated for each of the database entries, using statistical information that we gather from analysis of the humming database. Finally, a ranked list of melodies with respect to calculated similarity is returned to the user, as the output of the system.

1.2.4 A mid-level Representation Schema

In polyphonic domain, a true note transcription similar to what has been done in query by humming is not easy. Because of the complex structure of the polyphony, where an unknown number of sound sources are present, it is not easy to analyze and model note events from polyphonic audio. If an accurate way of performing true note transcription was available, it would be easy to systematically compare the audio input against the database elements. However, due to complex structure of polyphonic music audio, accurate analysis and modeling of sound units become very hard to achieve.

In this research we followed a mid level representation that tracks instantaneous changes of the tonality of the audio. Instead of trying to extract musical notes, the tonality is tracked by extracting frame level triad chord information from the audio for sequential modeling purposes.

1.2.5 Applying a tonal space in 3D (Spiral Array and its updated form) for tracking and labeling the harmonic movement of polyphonic audio

Throughout history, mathematicians and musicians have been largely interested in modeling pitches and their relations to each other for various reasons. Starting from the first construction of the tonnetz(the harmonic network) by Euler, mathematicians agressively debated the correctness of the forms of the harmonic models they proposed. Different implementations of these models have been updated and used for various reasons such as visually tracking harmonic movements in 3D, key finding, simiarity assestment. In this thesis, the Spiral Array array and its updated version will be used for calculating tonal centers that are generated from each consecutive audio frames. The tonal centers are labeled by the distance between the tonal center and pre-claimed space clusters of triad chords in the tonal object.

1.2.6 Sequential modeling and retrieval of polyphonic music

From the text transcriptions, sequential n-grams are created to track and model the tonal behavior of the polyphonic pieces. Constructed n-grams are then mixed with a Universal Background Model in order to compensate for different lengths of melodies. The collection of smoothed melodic n-grams creates a music database. While searching a polyphonic audio, its text transcription is ranked against the database entries using perplexity analysis. Perplexity, the relative entropy, will tell how likely a sequence of text transcription could be created by the smoothed n-grams in the database. From the perplexity scores, the system outputs an N-best list of relevant matches from the database. Here, also different forms of representations are used to suplement the plain tonal models. The supporting factor of two sub models which are pure harmonic and pure rhythmic will be discussed.

1.3 Related Work

This section summarizes related work in the literature about query-by humming and query by example pointing out general similarities and differences with respect to the work that is presented in this proposal.

Initial efforts on Query by Humming systems used note level decoding of humming input for pitch and duration contour representation. Ghias et al. [GLCS95] has been credited as one of the first to propose a QBH system that converts audio into melodic contour representation. The contour representation has been used widely by many QBH systems that followed. Ghias used autocorrelation for pitch tracking in order to encode the audio into a 3 level alphabet of pitch contours (U, D and S). McNab et al. [MSW$^+$96, MSWH00] introduced the duration concept into the melodic representation scheme. Jang et al. [CJ98, UCGN08] expanded the 3 level alphabets to one using the semitone (half step) distance measure, that gives information about the relative tonal change with positive and negative integer values of semitones. A positive integer represented an increase in pitch, and negative integers represented a decrease in the pitch contour. In this proposal, the representation that is chosen is similiar to the ones summarized above, a pitch contour representation of note transitions in semitones and duration ratios of consecutive notes. Note level decoding is also used for midi type representation where each located notes are assigned a midi number and a corresponding duration value [RRG99, PSB04].

Other than note level decoding, fixed size audio frame representation was also used which may avoid note transcription errors but can be computationally more expensive in the matching process [KNS$^+$99]. For retrieval, Dynamic Programming (DP) based ideas were evoked to find the closest match in a database. Dynamic Time Warping (DTW), a specific instance of DP for time series, is one of the most popular techniques in the MIR literature that has been used extensively in different forms [BR98, CJ98, KNS$^+$99, HD02, ZS03].

Pardo et al. [PSB04] and Jang et al. [CJ98] used sequential statistical modeling techniques such as HMM and CHMM's for representing audio elements in the database. A database melody is judged similar to the query if the HMM representation has a high likelihood of producing the query. Statistical modeling ideas are powerful for representing variability but, tend to be computationally expensive in terms of appropriate training.

Dannenberg et al. [DBT$^+$03] created a test-bed for comparing different retrieval systems in terms of performance for retrieval accuracy and computation time. They compared the search algorithms based on note-interval matching with dynamic programming, fixed-frame melodic contour matching with dynamic time warping, and a hidden Markov model. They reported that, contour matching with DTW and the HMM approach while extremely slow however is able to outperform all conventional faster methods. On the other hand, the note interval system was able to deliver similar results and was comparably faster than these two methods. Also, Ito et al. [IHSM04] compared different selected feature sets for DP based search algorithms. They introduced three different similarity measures: distance based similarity, quantization based similarity and fuzzy quantization based similarity. They concluded that the distance based similarity defined as absolute difference between two compared pitches, performed the best for retrieval experiments. Parker [Par05] presented a retrieval system using binary classification and adaptive boosting. After several iterations with boosting, the proposed system was able to outperform DTW algorithms with negligible increase in time.

Similar type of sequence alignment and comparison can be seen in genetic applications [DEKM98] employing statistical string matching techniques such as n-grams. However, the query input that we have in query by humming is rarely an exact substring of the original melody in the database because of the imperfect production of humming, so that a distance measure needs to be incorporated alongside the sequential alignment techniques.

With regards to QBH front-ends, Pauws [Pau02], in his Cuby-Hum, designed a new transcription technique that processes the input in terms of energy and pitch to detect note onsets and locations, quantizing the input frequencies into semitone representation. Pauws also used a DTW-ED approach for retrieval. Clarisse et al. [CML+02] first evaluated several existing transcription systems (such as Meldex, Pollastri, Autoscore, etc.) and, observing that they are not adequate for human level performance, constructed an Auditory Model Based transcription system for the front-end of QBH Systems. Also, inspired by the lack of robust humming audio transcription systems, Shih et al. [SNK02, SNK03, UCGN08] used HMM's in the front-end of the QBH system for statistically capturing the time location of notes in the input. Energy and mel frequency cepstral coefficient (MFCC) features were used to train the statistical models; for a fixed syllable input case (humming with /da/), 95% note segmentation accuracy was reported.

Not many other researchers have focused on accurate audio-to-symbol transcription in designing QBH systems; rather they focused more on different retrieval techniques. In this work, both the audio-to-symbol mapping problem and the retrieval problem are discussed in detail, and experimental results are reported.

For polyphonic music retrieval, MIR research mostly focused extracting musical features from the audio in order to understand the content of the music pieces. Some related topics are, note transcription, instrument detection, beat & rhythm tracking, chord & key detection, and etc. Using the extracted musical feauteres, researchers aimed at two main things. First, indexing of music files, and second retrieval with respect to the extracted feauture level content. In this proposal, our main point of interest is to find a way to accurately represent the audio files with symbols, to be used in retrieval. The mentioned representation should consider the tonal behaviour of the music piece that is under analysis, and should capture instantonaous changes in the tonal characteristics in time. In MIR literature, initial efforts of representing polyphpnic music is started with using a true audio to note transcription.

A considerable amount of research focused on the transcription of music audio signal to MIDI or piano roll type representations for accurate understanding of the tonal structure of a polyphonic tune. Numerous researchers modeled sound events in order to detect musical notes and their onset and offset times. Amongst them, Raphael [Rap02], Pertusa & Inesta [PI03], Smargadis & Brown [SB03], Ryynanen & Klapuri [RK05], and Poliner & Ellis [PE07] employed machine learning algorithms such as Hidden Markov Models, Bayesian networks, and Support Vector Machines, which performed well for mono-timbral transcription tasks, such as piano music transcription, where only one single instrument is allowed. These results are promising, however, their extension to a general solution for non-instrument specific polyphonic transcription remains in question.

Since accurate transcription of multi channel audio is not easy, a mid level representation of audio is mostly desired for tracking tonal behaviour of the music pieces for query by example problem. Recent research attempts in [TYW02, Mar, PBC$^+$02, SG08, UGCN07] showed that, different representation techniques such as extracting dominant melody or a chord progression from the music piece could be a feasible solution for polyphonic representation, since, harmonic structure of such variations tend to be stable even the expressive parameters of the performance change. DTW, n-grams or edit distance type of string matching techniques were used for retrieval. On the other hand, [ECM08, SG08] used chorma based features to directly represent music pieces, without any musical labeling, while simple cross correlation of chroma vectors were used for measuring similarity. Samuel also adopted delta chroma features in the representation which represents general movement in the harmonic structure of music pieces for more accurate representation and retrieval [SUN08].

The common solution to music audio matching and retrieval is to perform symbol-to-symbol comparison within a database to find the most similar, or the exact, matches to the input. Since the main features that are being used for the matching task are features (or symbols) that are extracted from the transcription process, the performance of the transcription directly impacts the performance of the matching and retrieval. In fact, retreival systems can be tolerant against some levels of uncertainty, so that the retrieval problem might be independent of the performance of accurate audio to symbol transcription.

Initial efforts in polyphonic music retrieval used MIDI transcriptions for modeling melodies. Doraisamy & Ruger [DR02] used MIDI transcriptions of musical pieces for comparing audio data; n-grams were built from different sets of features that were extracted from MIDI transcriptions, and the cosine rule was adopted for the ranked retrieval. Since MIDI is a text based representation of auido, where the note's exact locations (onsets and offsets), the timber and the intensitiy of the note is defined, the transcription from audio to symbols is not a challenge in here. However, the idea of statistical modeling of consecutive note events will be used in this paper, in a similiar fashion, that will be explained more in detail in Section 3.2.

Pickens et. al. [PMS+03] considered the query by example problem as a whole and proposed a general solution. They used existing polyphonic transcription systems in the literature to collect melodic note features from mono-timbral (piano only) music audio. The transcription was then mapped to a harmonic domain, namely a harmonic model that was designed for representing the auido with matrices corresponding to the distributions of the 24 lexical triads(three-note chords) and their transitions for the concurrent states. In later studies, Lavrenko & Pickens [LP03] used random fields to model polyphonic music pieces from MIDI files. Using random fields, they automatically induced new high level features from the melodies, such as consonant and dissonant chords, progressions and repetitions, to efficiently model polyphonic music information.

In this work, a similar strategy that involes a mid level tonal representation of audio and statistical tonal modeling for solving the query by example problem is used. The representation schema selected is slightly different from Pickens et. al.'s in that we prefer single dimensional representations of chord sequences, and the retrieval performance of the system is not directly limited by the performance of the initial audio-to-symbol transcription. Also we consider rhythm and harmonic progressions concurrently, however we were only able to see the performance increase with supporting harmonic progression models. Our aim for transcription from audio to meaningful symbols is a mid-level representation, which is independent of the exact note onsets and offsets, and also independent of the spectral effects of different musical instruments. Our optimization criterion in this work is not the transcription accuracy but the retrieval performance.

1.4 Outline of this thesis

This thesis is organized in two parts. Chapter 2 describes in detail our query by humming system, explicitly talking about first, our database corpus, its statistical analysis and findings, second, the symbolic representation that is used for humming queries, third, the statistical front end that achieves segmentation of musical notes in the query input and forth, the retrieval engine that uses fixed length information packages for robust matching and finally fifth, our experiments, results and discussion. Chapter 3 describes our system specifically designed for retrieval in the polyphonic domain. First, mid-level representation that is chosen for tracking tonal behaviour is introduced, and spiral array, a tonal space to track the tonal behaviour of a music piece with coordinates in three dimensional space is discussed. The Spiral is later updated with regards to the labeling process in order to eliminate repetitions that is introduced by the Spiral Array object for musical purposes. Later in the chapter, the algorithm that is used to map the coordinates into musical symbols is explained and sequentiontal modeling of polyphonic music is discussed. Different forms of representation is also discussed with supporting pure time related and pure tonal related models. At the end of the chapter, retrieval experiments are presented with results and their detailed discussion. The thesis ends with conclusion, with possible future directions on both proposed systems.

Chapter 2

Query by Humming

Content-based multimedia data retrieval is a developing research area. Integrating natural interactions with multimedia databases is a critical component of these kinds of efforts. For Music information Retrieval (MIR), using humming, a natural human activity, for querying data is one of ways for facilitating such interactions. Mode of interaction in music databases requires audio information retrieval techniques to be developed for mapping the human humming waveforms to pitch number strings representing the pitch and rhythm contours of the underlying melody. A query engine then needs to be developed in order to search for the converted symbols in the database. The query engine should be precise and robust to interuser variability and uncertainty in query formulation.

A QBH system usually has two components: a) a segmentation (transcription) engine, which maps the audio signal into a symbolic representation, and b) a retrieval engine, which compares the symbolic representation to those of the database elements for searching relevant matches (Figure 2.1). Both the transcription and the retrieval engine needs to be robust against user dependent variability, namely the uncertainty. Moreover, the retrieval engine also needs to compansate for the errors that is generated by the transcription engine, such as insertions and deletions of notes, alongside melodic errors that is generated by the user. Therefore the transcription engine needs to operate as accurately as possible since, its performance directly effects the performance of the retrieval calculations.

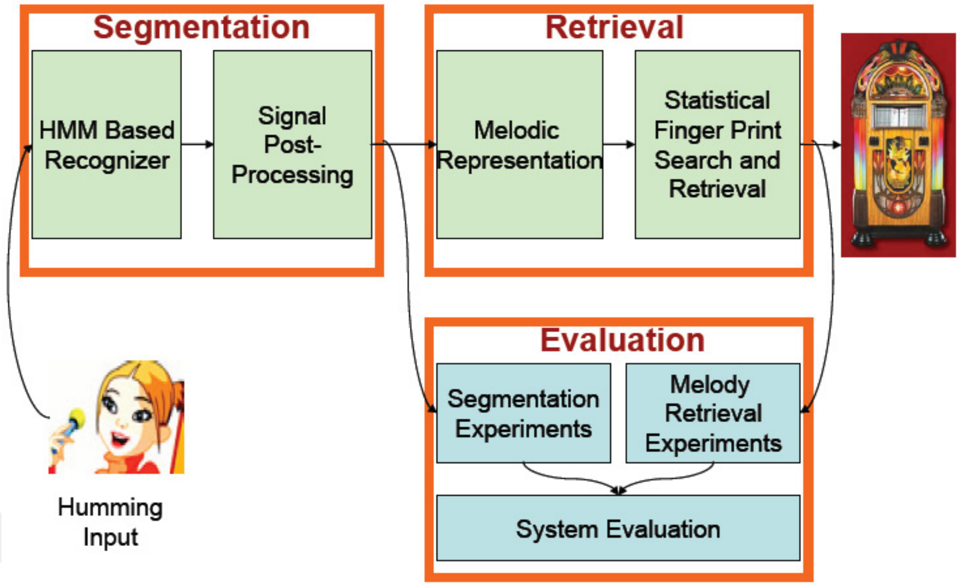


Figure 2.1: Proposed Query-by-Humming System: Segmentation, Retrieval, Evaluation

Our proposed statistical approach to humming recognition aims at providing note-level decoding using statistical models (we favor hidden Markov models or HMMs) of audio features representing melodies. Since the approach is data driven, it promises robustness in terms of handling human variability in humming. Conceptually, the approach tries to mimic a humans perceptual processing of humming as opposed to attempting to model the production of humming. Such statistical approaches have had great success in automatic speech recognition and can be adopted and extended to recognize human humming and singing [SNK02]. In order to achieve this, a comprehensive humming database needs to be developed that captures and represents the variable degrees of uncertainty that can be expected by the frontend of the QBH system.

Outline of the chapter is as follows. In Section 1, the data collection experiments are discussed. In this section, the outcome of the statistical analysis are also reported and discussed. In Section 2, the first component of our proposed QBH system, the transcription engine is presented. After that, in section 3 the statistical finger prints are discussed. This section also explains how the retrieval calculations are performed by the guidelines that are deducted from the analysis of the collected data. Finally in Section 4 the retrieval experiments and their results are discussed in detail.

2.1 Humming Data Collection and Analysis

Advances in music retrieval research greatly depend on appropriate database resources and their meaningful organization. In this section we describe data collection efforts related to the design of query-by-humming (QBH) systems. We also provide a statistical analysis for categorizing the collected data, especially focusing on intersubject variability issues. In total, 100 people participated in our experiment, resulting in around 2000 humming samples drawn from a predefined melody list consisting of 22 different well-known music pieces and over 500 samples of melodies that were chosen spontaneously by our subjects. The data from each subject were compared to the expected melody features, and an objective measure was derived to quantify the statistical deviation from the baseline. The results showed that the uncertainty in human humming varies depending on the musical structure of the melodies and the musical background of the subjects. Such details are important for designing robust QBH systems.

Our goal in this study is to create a humming database that includes samples by a cross-section of people with various musical backgrounds in order to make statistical assessments of intersubject variability and uncertainty in the collected data. Our research contributes to the community by providing a publicly available database of human humming, one of the first efforts of its kind. The collected data will be used in two main tasks: a) estimating important parameters of the front end transcription engine, and b)for testing the accuracy of the retrieval engine.

Building a statistical system that performs pitch-and-time-information-based retrieval from a humming sample has been shown to be feasible [SNK02]. However, since the quality of the input depends largely on the user, and includes high rates of variability and uncertainty, a key challenge is achieving robust performance under such conditions. Next, we will discuss our hypothesis on the sources of uncertainty in humming performance. Since our approach is based on statistical pattern recognition, it is critical that the test and training data adequately represent the kinds of variability expected.

2.1.1 Hypothesis

The data collection design was based on certain hypotheses regarding the dimensions of user variability.We hypothesize that the main factors contributing to humming variability include the musical structure of the melodies being hummed, the subjects familiarity with the song, and the subjects musical background and that these effects can be modeled in an objective fashion using the audio signal features.

Musical Structure

The succession of notes and the rhythm of a melody are the features that greatly influence how well a human can faithfully reproduce them through humming. Some melodies possess a very complex musical structure such as difficult note transitions and complex rhythmic structures that make them difficult to hum. When we create a database, one criterion is to populate it with samples reflecting a range of musical structure complexity. In this regard, the note succession as notated in the score of the melodies was the information we used to determine the musical complexity.

Pitch range is an important factor affecting the difficulty of humming a melody. We measured the pitch range of the songs according to two statistics: the difference between the highest and the lowest note of the melody and, more importantly, the largest semitone differential (interval) between any two consecutive notes. For example, two of the wellknown melodies we asked our subjects to hum “Happy Birthday” and “Itsy Bitsy Spide” have different musical characteristics according to those measures. The range of notes in “Happy Birthday” spans one full octave (12 semitones), while the range in “Itsy Bitsy Spider” is only 5 notes (7 semitones). Moreover, the highest absolute pitch change between two consecutive notes in “Happy Birthday” is again 12 semitones, while the same quantity is only 4 semitones in “Itsy Bitsy Spider.” On the other hand, one of the melodies in our melody list was the “United States National Anthem.” Its note collection spans 19 semitones, and the highest differential between two consecutive notes is 16 semitones, not an easy interval for nonprofessionals to sing accurately. If we want to compare these three songs, we can speculate that the average performance of the humming of “Itsy Bitsy Spider” will be better than the performance of the humming of “Happy Birthday” or the “United States National Anthem.”

Apart from pitch range, difficulty can also be a function of “perceived closenes” of intervals in terms of fractions between pitch frequencies. For example, the interval of 7 semitones (corresponding to a perfect fifth and approximately a frequency ratio of 2:3) is a simple relationship to make, and thus sing, whereas an interval of 6 semitones (corresponding to an augmented fourth or diminished fifth and approximately a frequency ratio of 5:7), although closer in terms of frequency, is usually more difficult to sing. Hence it is important to incorporate information about the type of intervals.

Familiarity

The quality of the reproduced melody (singing or humming) also depends on the subjects familiarity with the specific melody. The less familiar the subject is with the melody, the higher the expected uncertainty. On the other hand, even though a melody may be very well known, it does not mean that it would be hummed perfectly, as evidenced by many performances at karaoke bars. Therefore, we prepared a list of well-known pieces ("Happy Birthday" "Take Me Out to the Ball Game," etc.) and nursery rhymes ("Itsy Bitsy Spider," "Twinkle Twinkle Little Star," etc.) and asked our subjects to rate their familiarity with each melody. In her studies about relevance assessment, Uitdenbogerd believed that it was a very difficult task for users to compare and process unknown pieces of music [Uit03]. This result also supports our hypothesis that the humming performance will be better when our subjects hum the melodies with which they are more familiar.

Musical Background

We can expect musically trained subjects to hum the melodies we ask with a higher accuracy, while musically nontrained subjects are less likely to hum the melodies with the same degrees of accuracy. By musically trained we mean that the subject has had some formal music training, for example through classes such as diction, instrumental instruction, or singing lessons. Whether or not the instruction is related to singing, even a brief period of instrumental training affects ones musical intuition. On the other hand, we also know that musical intuition is a basic cognitive ability that some nontrained subjects may already possess. We, in fact, experienced very accurate humming from some nontrained subjects in our database. Hence another goal of the data acquisition was to sample subjects of varied skills.

2.1.2 Experimental Methodology

Given the aforementioned goals, the actual corpus creation was done according to the following procedure.

Subject information

Since our project does not target a specific kind of user population, we encouraged everyone to participate in our humming database collection experiment. However, in order to enable the performance of informed statistical analysis, we asked our subjects to fill out a form that requested information about their age, gender, and linguistic and musical background. The personal identity of the subjects was not documented in the database. Most of the participants were university students who were compensated for their participation per institutional review board approval for human subjects.

Melody List and Subjective Familiarity Rating

We prepared a list of 22 melodies that included folk songs, nursery rhymes, and classical pieces. These melodies were categorized by their musical structure, in total covering most of the possible note intervals in their original score (perfects, majors, minors). Table 2.1 shows the number of intervals we covered for each interval type in both ascending and descending format. The melody set only lacks a major seventh interval, which corresponds to an 11-semitone transition.

The melodies containing large interval leaps were assumed to be the more complex and difficult melodies (“United States of America National Anthem” “Take Me Out to the Ball Game” “Happy Birthday”), and those containing smaller intervals were assumed to be the less complex melodies (“Twinkle Twinkle Little Star,” “Itsy Bitsy Spider,” “London Bridge”...). The full melody list used for this corpus is available online at the project Web page (http://sail.usc.edu/music/). These melodies were randomly listed on the same form where we asked our subjects to give their personal background information. The form template is also available online (http://sail.usc.edu/music/).

At this stage we asked our subjects to rate their familiarity with each melody using a scale of 1 to 5 after hearing the melodies played from the computer as MIDI files, with 5

Table 2.1: Interval Types and Occurances in the Selected Target Melody Database

		Frequency		
Semi-tones	Interval Type	Ascending	Descending	Total
0	Perfect Unison	199		199
1	minor 2nd	43	39	82
2	Major 2nd	185	48	233
3	minor 3rd	27	43	70
4	Major 3rd	15	33	48
5	Perfect 4th	22	14	36
6	Aug 4th/dim fifth	2	-	2
7	Perfect 5th	9	10	19
8	minor 6th	4	4	8
9	Major 6th	7	4	11
10	minor 7th	2	-	2
11	Major 7th	-	-	-
12	Perfect Octave	4	-	4

being the highest level of familiarity. Subjects used 1 for rating melodies that they were unable to recognize from the MIDI files.

During the rating process we asked our participants to disregard details regarding the lyrics and the name of the melody, as we believe that the tune itself is the most important feature.

Humming Query

After the familiarity rating process we picked ten melodies that were rated highest by the subjects. We asked them to sing each of these melodies twice using . . .da, da, da. . ., a stop consonant-vowel syllable. Any choice of a stop consonant sound was allowed, which the subject was most confortable with.

Equipment and Environment

A digital recorder is a convenient way of recording audio data. We used a Marantz PMD690, a digital recorder, which provides a convenient way to store the data to flash memory cards. The ready-to-process humming samples were transferred to a computer hard disk and the data were backed up on CDRs. Martel, a tie-clip electret condenser microphone, is preferred for its built-in filters that lower the ambient noise level. The entire experiment was performed in a quiet office room environment to keep the data as clean as possible.

2.1.3 Data

In total, we have acquired thus far a humming database from 100 participants whose musical training varied from none to 25+ years of professional piano performance. These people were mostly college students over 18 years of age and from different countries. Each subject performed 20 humming pieces from the predefined melody list and 6 humming pieces of their own choice, giving us a total of over 2500 samples. This humming database is being made available online at our Web site and will be completely open source. The instructions for accessing the database will be posted at the Web site (http://sail.usc.edu/music/). For convenient access and ease of use, the database needs to be well organized. We gave unique file names to each humming sample. These file names include a unique numerical ID for each subject, the ID of the melody that was hummed, and the personal information of the subject (gender, age, and level of musical training). We also included an objective measure of uncertainty at the end (Sects. 5 and 6). The file format is as shown:

$$txx(a/b)(+/-)pyyy(m/f)zzww \tag{2.1}$$

where xx is an integer value that gives the track number of the song from the melody list being hummed, (a/b) specifies whether the sample is the first or second performance, (+/-) indicates if the subject is musically trained, yyy stands for the personal ID number, (m/ f) gives the gender of the subject, and zz tells us the persons age. ww is a float number that shows the average error per note transition in semitones, which does not necessarily correspond to the quality of humming.

2.1.4 Data Analysis

One of the main goals of this study is to implement a way to quantify the variability and uncertainty that appear in the humming data. We need to distinguish between good and bad humming, not only subjectively but also objectively from the viewpoint of automatic processing. If a person is musically trained and listens to the humming samples that we collected, s/he can easily make a subjective decision about the quality of the piece with respect to the (expected) original. However, this is not the case with which we are primarily concerned.

For objective testing, we analyze the data with a signal processing freeware software named PRAAT and retrieve information about the pitch and timing of the sound waves for each of the notes that the subject produced by humming. Each humming note is segmented manually, and for each segmented part we extract the frequency values with the help of Praats signal processing tools. Rather than the absolute values of the notes themselves, we analyze the relative pitch difference (RPD) between two consecutive notes [1, 6]. The pitch information we obtained allows us to quantify the pitch difference at the semitone level by using the theoretical distribution of semitones in an octave. RPD is calculated using the following formula:

$$RPD = \frac{log(f_k) - log(f_{k+1})}{\sqrt[12]{2}} \tag{2.2}$$

where f is the f0, the percieved frequency and k is the index of the transition in the note sequence. In this study, we define humming error as numerical semitone level difference between the hummed note transition and the target note transition. User's humming error in one note transition is calculated by:

$$Error = \frac{OT - RPD}{OT} \tag{2.3}$$

where RPD is already defined as the relative pitch difference between two consecutive hummed notes, and the OT is the original transition which the subject tried to hum. Error metric tells how far a performed transition is away from the original transition and since both the denominator and numerator are in semitones, the metric has no units.

2.1.5 Performance Comparison in Key Points

During data collection we observed varying performance levels at different parts of each melody. The most common parts where subjects made the most significant errors are the wide range note transitions, the first couple of notes of each melody where subjects make key calibrations, and some specific intervals defined as inharmonic such as augmented/ diminished intervals.

Wide Range Note Transitions

The humming sample as a whole ismost affected by large interval leaps in the original melody.While large interval transitions are difficult for nontrained subjects to sing accurately, the same is not true for musically trained people. A musically trained subject will not necessarily hum the melody perfectly. However, their performance at these challenging transitions can be expected to be more precise.

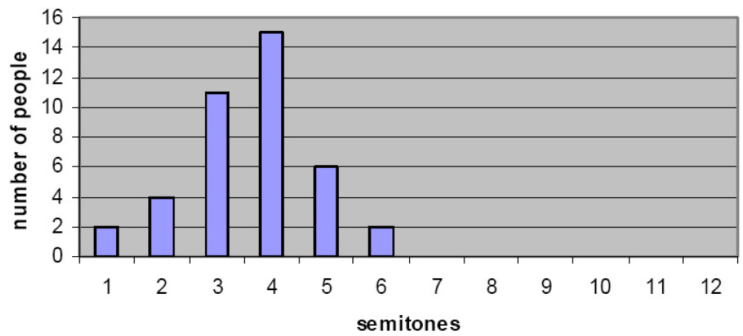


Figure 2.2: Humming performance of the selected control group for the song “Itsy Bitsy Spider” (first two phrases) at the highest semitone level difference.

Figure 2.2 shows the distribution of the actual intervals sung by 20 randomly selected subjects at the point of the largest interval leap in “Itsy Bitsy Spider.” Each subject hummed the melody twice. This particular melody, shown in Figure 2.3, is one of the easiest melodies in our database, having a maximum note-to-note transition interval of “4” semitones (marked by $(*)$ in the score).

Figure 2.3: “Its Bitsy Spider” melody score

Ten of the subjects in this particular test group are musically trained, so we analyze a total of 20 (each participant hummed a melody twice) samples from musically trained subjects and 20 samples from untrained subjects.

As seen from the figure, the mode (highest frequency) of the performance for this interval is 4, the actual value. Fifteen out of 40 samples show accurate singing of this interval, and 10 of these accurate samples are performed by musically trained people.

The average absolute error made by musically trained subjects in humming that interval transition is calculated to be 0.63 semitones, while this value is 1.29 semitones for non-trained subjects. As expected, the largest interval sung by musically trained subjects is 104.8% better than the performance of nontrained subjects with respect to the decrease in error rate.

Figure 2.4: “Happy Birthday” melody score

To further investigate, we then analyze the humming samples performed by the same control group for the melody “Happy Birthday,” which is shown in Figure 2.4. The largest interval skip in “Happy Birthday” is 12 semitones (one octave is labeled $(*)$), which is a relatively difficult melodic leap for untrained subjects. “Happy Birthday” is one of the examples containing a large interval in our predefined melody list. Figure 2.5 shows the performance distribution of the previous control group for the humming of “Happy Birthday.”

The mode for the singing of the largest interval is 12, the size of the largest interval in “Happy Birthday.” Fifteen out of 40 samples are accurate in reproducing this particular interval, and 11 of these are by musically trained subjects. The average absolute error calculated for musically trained subjects is 0.845 semitones and, the average absolute error in nontrained subjects’ performance is 1.963 semitones. These values show that musically trained subjects performed 132.3% better than the nontrained subjects in singing the largest interval in “Happy Birthday.”

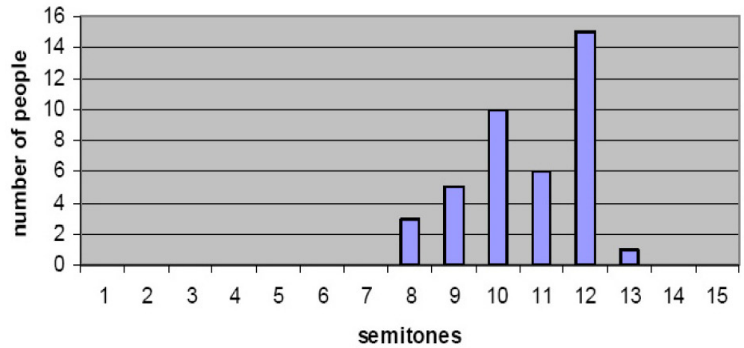


Figure 2.5: Humming performance of the selected control group for “Happy birthday” at the highest semitone level difference

A simple factor analysis of variance (ANOVA) for the songs “Itsy Bitsy Spider” and “Happy Birthday” indicates that the effect of musical training on the accurate singing of the largest intervals is significant. (“Itsy Bitsy Spider” $\rightarrow F(1, 39) = 8.747$ $p = 0.005$; “Happy Birthday” $\rightarrow F(1, 39) = 10.630$ $p = 0.002$.)

Key Calibration

Subjects experienced key calibration problems at the start of each humming, and they performed with higher error levels at the beginning of the melody. This may be because, for a certain time at the beginning, subjects try to adjust their humming to the key they have in their mind, and this transition period results in unexpected levels of error in the fundamental frequency contour. This orientation period is mostly obvious in nontrained subjects.

To investigate this hypothesis, we analyze the first interval of each humming sample and compare the performance of subjects at the same interval in later parts of the same melody.

Consider the melody “London Bridge” shown in Figure 2.6. As seen from Table 2.2, the analysis shows that, for “London Bridge,” the error value calculated for the performance of the first interval of the melody (a major second interval or 2 semitones labeled $(*)$ in the score) is 0.542 semitones, and the error value for the performance of the same interval that occurred later (randomly selected from major second intervals labeled $(\%)$) in the same melody is calculated to be 0.138 semitones. The performance improvement is a remarkable 74.5%.

Figure 2.6: “London Bridge is Falling Down” melody score

Figure 2.7: “Did You Ever See a Lassie” melody score

Table 2.2: Calculated errors at various locations vs interval kinds

	Begining	Elswhere	Improvement
2 semitones: Major 2nd Interval	0.542	0.138	74.5%
4 semitones: Major 3rd Interval	0.773	0.367	52.5%

We present another example, “Did You Ever See a Lassie,” shown in Figure 2.7. Because of the key calibration problem, subjects performed 52.5% better at the minor third intervals (labeled (%) that are in the melody as compared to the one at the beginning (labeled $(*)$).

A simple factor analysis of variance (ANOVA) for the songs “London Bridge” and “Did You Ever See A Lassie” indicates that the effect of key calibration at the beginning of the humming is significant. (“London Bridge” → $F(1, 47) = 12.800$ $p = 0.001$; “Did You Ever See A Lassie” → $F(1, 39) = 10.473$ $p = 0.002$) The results are summarized in Table 2.2.

Special Intervals

We also had a chance to observe the effect of dissonance, which refers to the perceptual quality of sounds that seem “unstable” and have a need to resolve to “stable” sounds. As discussed in Section 2.1.1 it is hypothetically more difficult to sing an augmented fourth interval (6 semitones) versus the wider perfect fifth interval (7 semitones).

Figure 2.8: “Twinkle Twinkle Little Star” melody score

To investigate this, the performance of a perfect fourth (5 semitones, frequency ratio approximately 3:4), an augmented fourth (6 semitones), and a perfect fifth interval (7 semitones) using humming samples from a control group of 20 subjects are analyzed, and average error values are calculated for each interval. For statistics on the singing of the perfect fourth (labeled (%)) and perfect fifth intervals (labeled $(*)$, we analyze

Figure 2.9: “Maria” melody score

the song “Twinkle Twinkle Little Star” (Figure˜twinkle), and for the augmented fourth interval (labeled $\langle @ \rangle$) we analyze the song “Maria” from “the West Side Story” (Figure 2.10).

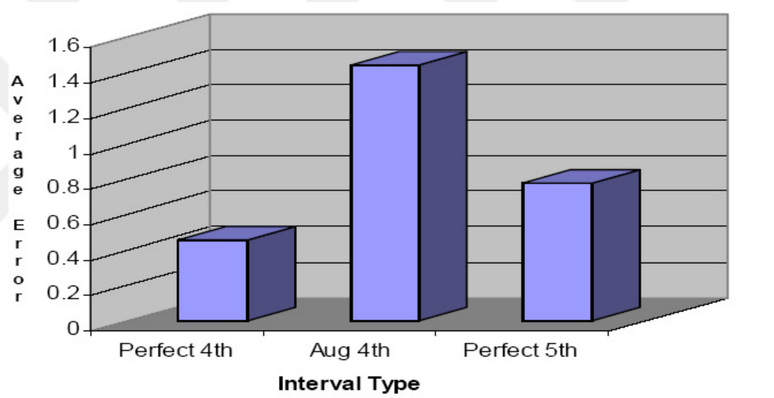


Figure 2.10: Comparison of the average error calculated with the interval type: Perfect 4th, Augmented 4th and Perfect 5th

A simple factor analysis of variance (ANOVA) for the singing of the perfect fourth, augmented fourth, and perfect fifth intervals indicates that the effect of dissonance on the calculated error per interval is significant. (“Perfect 4th and 5th Intervals and Augmented 4th intervals” $\rightarrow$ $F(1,47) = 13.700$ $p = 0.001$.)

2.1.6 Performance Comparison Across the Whole Melody

In the melody "Itsy Bitsy Spider" (Figure 2.3), there are 24 notes and 23 transitions. For each interval, Figure 2.11 compares the interval sung by an untrained subject with that occurring in the original piece.

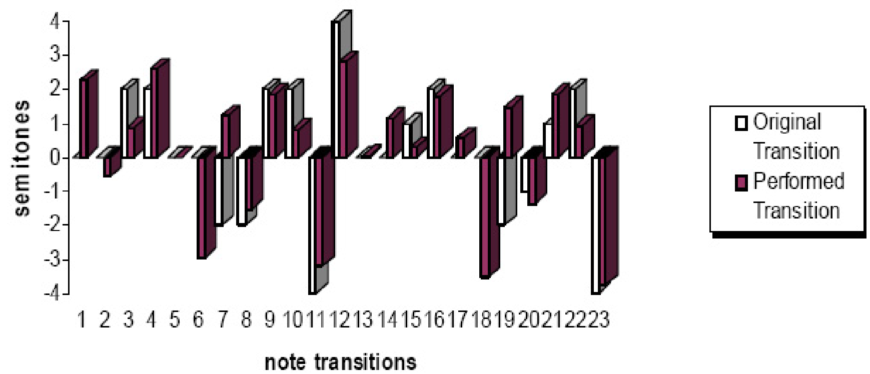


Figure 2.11: Comparison of humming data with the base melody at each note transition for nontrained subjects (shown for "Itsy Bitsy Spider")

For each interval transition we calculate the error between the observed data and the original expected values in semitones. The sum of all these values gives us a quantity that serves as an indicator of the quality of this particular humming sample. In the case shown in Figure 2.11, this subject performs with an average error of 1.16 semitones per interval. Figure 2.12 compares a musically trained subjects humming with the original melody. The analysis shows that the average error in this musically trained subjects humming is 0.28 semitones per transition, expectedly lower than the error that we calculated in the nontrained subjects humming.

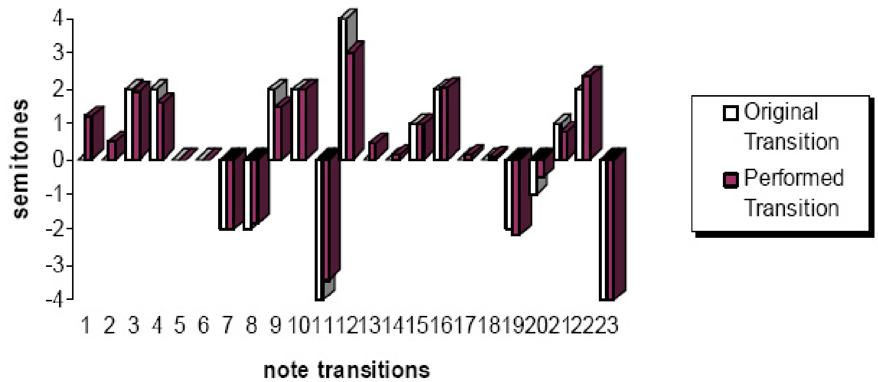


Figure 2.12: Comparison of humming data with the base melody at each note transition for trained subjects (shown for “Itsy Bitsy Spide”)

2.1.7 Performance Comparison over Musical Background of the User and Musical Structure

From Table 2.3 one can see that the uncertainty in the musically trained subjects humming is less than that in the nontrained subjects humming of the same song.

Table 2.3: Average Error values in semitones in trained and non-trained subject’s humming data for the melodies “Itsy Bitsy Spider” and “Happy Birthday”.

	Itsy Bitsy Spider	Happy Birthday
trained	0.43	0.47
non-trained	0.63	0.70
All-subjects	0.53	0.58

The average error value in the humming of the musically trained subjects in our control group is 0.43 semitones per transition in the melody “Itsy Bitsy Spider” The average error value for the nontrained subjects is 0.63 semitones per transition.

“Happy Birthday” previously hypothesized to be a more difficult melody to hum because of its intervals and range, produces the expected results as well. The average error for trained subjects is calculated to be 0.47 semitones per note transition, which is larger than the value of the same subjects performed while humming "Itsy Bitsy Spider" and the average error that is calculated for the nontrained subjects is 0.70, which is also larger than the error for the same subjects humming "Itsy Bitsy Spider."

We conclude that one can expect larger error values in the humming of musically nontrained subjects compared to that of musically trained subjects, as explained in Section 2.1.1. The ANOVA analysis shows that the effect of musical background is also significant for humming quality. ("Itsy Bitsy Spider"→$F(1, 39) = 12.062$, $p = 0.001$; "Happy Birthday" → $F(1, 39) = 8.646$, $p = 0.006$.) In addition, we also expect more uncertainty when the hummed melody contains intervals that are difficult to sing as previously discussed and explained in Section 2.1.1. The ANOVA analysis of humming performance of "Itsy Bitsy Spider" and "Happy Birthday" shows that the effect of musical structure is also significant. ($F(1, 79) = 5.91$, $p = 0.017$.)

2.1.8 Conclusion

In this section, we discussed our corpus creation for designing user-centric front-ends for Query by Humming Systems. We first created a list that included the melodies to be hummed by the subjects. This list was created based on specific underlying goals. We included some melodies that are deemed difficult to hum as well as some familiar and less-complex nursery rhymes. The experimenter decided what songs a subject was going to hum with an initial assessment of the musical background of the subject and the familiarity ratings that the subject assigned at the beginning of the experiment. After collecting data for this specific melody list, the subjects were asked to hum some self-selected melodies not necessarily in the original list. The data was organized by subject details and objective quality measures and is being made available to the research community. We performed preliminary analysis of the data and tried to implement a way to quantify the uncertainty in the humming performance of our subjects, with the help of signal processing tools and knowledge of the physical challenges in humming large intervals. We believe that this procedure increases the validity of the data in our database.

2.2 Front End Recognizer

In this section, the problem of melodic transcription and representation in the front end of QBH Systems is discussed. Each humming unit represents a single note in a melody. One of the main goals of the front-end system is to provide correct segmentation of the humming notes one from another. After segmentation, the pitch and duration values for each segmented note can then be extracted. We implemented a two-stage note transcription process comprising of first, an algorithm that uses a speech recognizer for note segmentation and, then signal processing techniques for precisely extracting pitch and duration information. This transcription process allows for multiple humming syllable types in various forms, which is typical of real world humming data, and incorporates post-processing using pitch and energy features to enhance system performance.

Here, we provide justification for the use of a speech recognizer for note segmentation. Humming can be defined as the reproduction of a melody without the words. Instead of the lyrics, fixed syllables such as /Da/, /Na/, /Ta/, /Ra/ and their different pronunciations are used. The main difference in salient feature sets between spontaneous speech syllables and humming units is the pitch and duration structure. The duration of the notes in the humming sequence (depending mainly on the tempo of the humming) is more likely to be longer than the syllables used in spontaneous speech. On the other hand, since each humming unit represents a music note, the pitch in the hummed notes tends to be more stable, while pitch in spontaneous speech may vary with expressiveness. Since the duration and the pitch contour characteristics are the only differences, and these differences do not significantly affect speech recognition performance, humming can be considered ”meaningless” (non-lexical) speech for the purposes of humming unit (note) segmentation. This suggests that adopting an automatic speech recognizer for locating hummed notes in the input query could be a feasible approach for segmentation. Admittedly, there exist alternative methods to using a speech recognizer to detect note onsets, such as through analysis of the short-term energy profile. But these methods are not robust as standalone techniques.

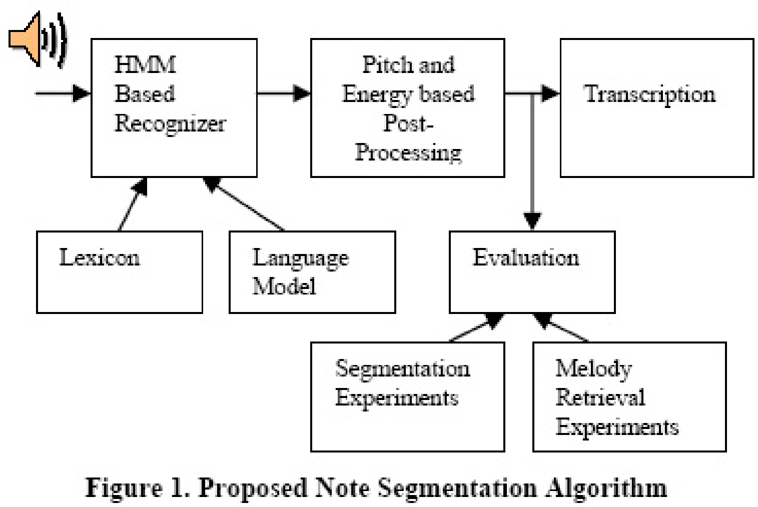


Figure 2.13: Note Segmentation Algorithm

Figure 2.13 shows a diagram of our note transcription system. In order to perform correct note segmentation, we design a robust Hidden Markov Model (HMM) based speech recognizer. The speech recognizer's task is to perform pre-segmentation by using statistical time-series note models. In the recognizer, we create 4 general models for different types of humming syllables that can be expected in humming audio. Our lexicon (dictionary used in the recognizer) groups the humming syllables into main categories with respect to their linguistic structures for accurate recognition. We train the generic humming models with available speech data corpora, and a “language” (note sequence) model with transcribed real world humming data. The output of a phoneme-level speech recognizer is post processed with energy and pitch features to detect the humming note boundaries. It is then segmented into audio chunks that correspond to the detected humming notes.

Finally, we evaluate the segmentation performance of the system with precision and recall (F measure) analysis. We also use the output of the recognizer for melody retrieval experiments in order to investigate how segmentation errors impact the performance of QBH systems.

2.2.1 Note Segmentation and Post Processing

In order to emulate the way humans perceive and reproduce music in a QBH system, the front end needs to robustly extract pitch and duration representations of the hummed notes. This requires accurate segmentation of the hummed notes in the query audio. In this work, an HMM-based speech recognizer, in conjunction with targeted post processing of its output, is used for note segmentation. The recognizer output is ultimately converted to meaningful musical symbols using the same representation as that used for the original melody database entries.

Automatic Segmentation Setup

One way of segmenting hummed notes is to use the human knowledge of how users are likely to hum. Users often use syllables, such as the “La” sound in order to hum a melody. To take advantage of this additional knowledge we built an HMM-based speech recognition system for note segmentation using mel frequency cepstral coefficients (MFCC) [UNC+06]. The system is built on the ARPA Wall Street Journal task

[PB92] of continuous English speech (50-phoneme set). Based on this speech recognition system we defined 4 generic syllable model types, denoted by /Da/, /Ta/, /Na/, /Ra/ in the lexicon, where each model represents a single type of consonant that is expected at the beginning of a hummed note (/Da/: voiced stops (b, c, d, g), /Ta/: unvoiced stops (p, t, k), /Na/: nasals (m, n), and /Ra/: liquids (l, r)).The lexicon also includes the different types of vowels (AA, AH, IH, AE) that are expected to follow the consonant to form the hummed syllable. The four generic syllable models provided in the dictionary aim at allowing all different types of musical syllables that can be used by the human subjects regardless of the consonant at the beginning of the humming syllable or the vowel following. The lexicon that is supplied to the recognizer is summarized in Table 2.4.

Table 2.4: Lexicon: Bootstrap models with their phoneme structures

Model	Consonant	Phoneme
DA (voiced stops)	D	IX
	D	AE
	D	IH
	...	...
TA (unvoiced stops)	T	IX
	T	AE
	...	...
RA (Liquids)	R	IX
	R	AE
	...	...
LA	L	IX
	L	AE
	...	...
NA (Nasals)	N	IX
	N	AE
	N	IH
	...	...

Figure 2.14 below shows a typical humming input, with the recognizer's output labeled on the waveform as note boundaries. As seen from the spectrum, each note is labeled with one of the note models that is supplied in the lexicon.

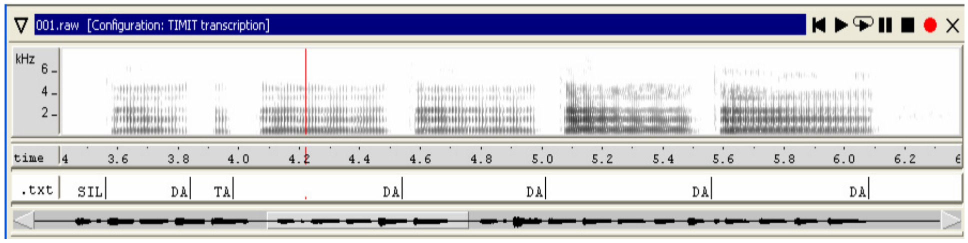


Figure 2.14: Visualization of the recognizer output: detected notes are labeled under the spectrogram as humming syllable models - {/DA/, /TA/, /NA/, /RA/, or /SIL/.}

Segmentation Errors and Post Processing

The speech-recognition based front-end segmentation procedure's output can contain errors, such as insertions and deletions of notes. Since duration and pitch information is derived from the segmented notes, ensuring the accuracy of the extracted note onsets and offsets is important. Figure 2.15 shows an example of a note deletion error, where 2 notes between the vertical lines are detected as a single note. Figure 2.16 shows an insertion error at the end of the first detected /DA/.

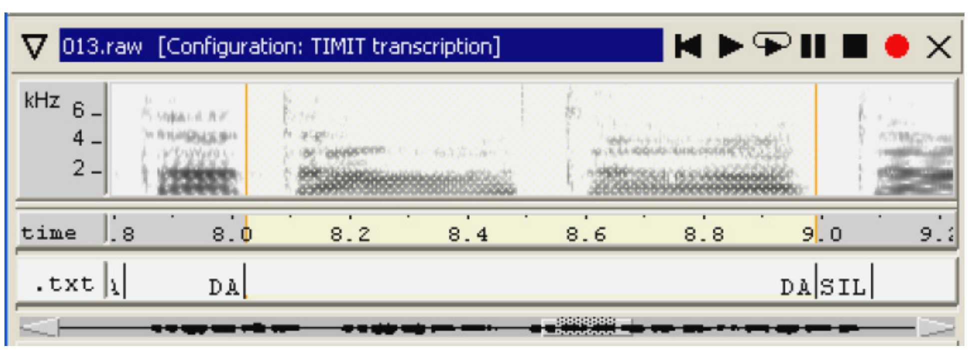


Figure 2.15: A deletion error in the recognizer's output: missing boundary between two notes in the center (between the vertical lines.)

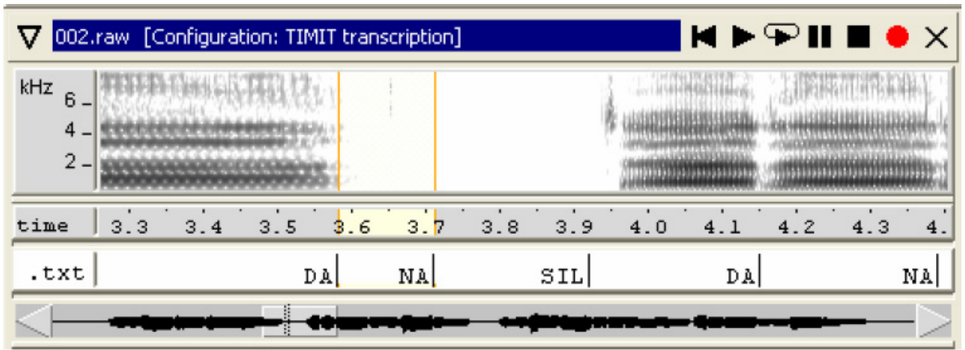


Figure 2.16: An insertion error in the recognizer's output: extra note /Na/ is detected after the first /DA/, should be /SIL/.

The note segmentation accuracy directly affects the performance of the retrieval system. If the recognizer fails to segment two or more hummed notes from one another, the feature extraction mechanism will consider these non-segmented units as one single note, leading to erroneous extraction of note pitch and duration. These insertions and deletions will in turn cause retrieval mismatches.

In order to detect common segmentation errors, and to correct them when possible, the output of the recognizer is post processed using heuristics based on energy and pitch information. As mentioned earlier, the recognizer uses MFCC's as the feature set. Incorporating the additional energy and pitch information as a new feature set at this point can be useful for correcting such segmentation errors introduced by the recognizer, whose MFCC based models have been trained with speech data. Since standard energy extraction algorithms are more accurate than pitch extraction algorithms, energy analysis is applied first. Pitch analysis is used as a final step for correcting segmentation errors.

Short-term Energy Analysis

We designed our energy analysis post-processing step specifically for correcting deletion errors. We compare the extracted energy feature with a threshold value derived from a small held-out set to introduce new note boundaries that are undetected by the recognizer. For each segmented humming note at the recognizer's output, the signal is windowed into frames of 20ms, with a shift of 10ms, which creates a 50% overlap between consecutive analysis frames. For each frame k, the short-term energy is calculated as

$$E_k = \sum_{m=1}^{N} y(m)^2 \tag{2.4}$$

where N is the number of samples in an analysis frame (sampling rate$\times$0.02). For the energy sequence E, an adaptive threshold value τ_e is defined by the product of the median value of E' (non-zero elements of the Energy sequence E), and a constant α that is calculated from a development set. Values in E, greater than the threshold are quantized to 1, and the values smaller than the threshold are quantized to 0. A note onset is detected if a 1-to-0 transition is followed by a 0-to-1 transition. The onset is positioned at the 0-to-1 transition point. The offset of the first note is given by the time of the 1-to-0 transition. The procedure can be seen in Figure 2.17.

Pitch Analysis

Alongside energy, the pitch vector ignored by both the recognizer and the energy analysis, can also be helpful in finding deletion errors. Since each segmented audio chunk corresponds to a humming note, where the pitch vector tends to stay the same, the analysis of pitch vector extracted for consecutive frames can be used to detect immediate changes within the corresponding note.

Pitch tracking for the hummed signal is performed using a standard pitch detection algorithm with the PRAAT software using autocorrelation method. The extracted pitch was stored in a sequence P. For each segment, the gradient of P was calculated and compared to a threshold value τ_p, which was estimated using a development set. Values that fall between the threshold regions are quantized to 0 and the rest are set to 1. An onset is determined when a 0-to-1 transition is followed by a 1-to-0 transition. The

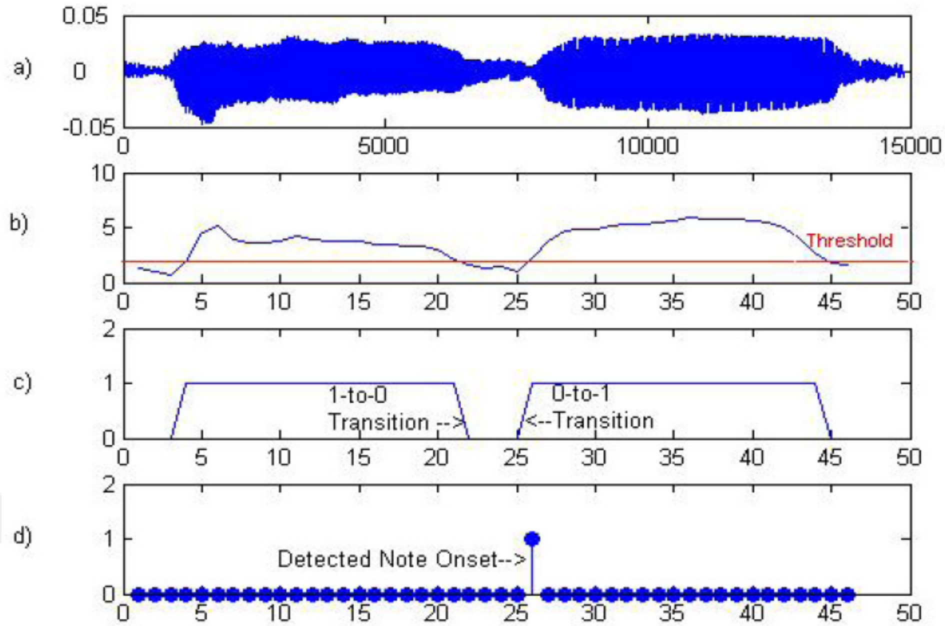


Figure 2.17: Example of deletion correction using energy-based threshold filtering: a) two humming syllables which the recognizer considered to be one; b) corresponding energy sequence and local energy threshold; c) quantization with respect to the threshold value; and, d) correct detected note onset.

onset is positioned at the time of the 1-to-0 transition. The procedure is illustrated in Figure 2.18.

Segmentation Evaluation

Reference transcriptions for the hummed samples are created manually by an experienced music student. Each manual transcription is compared to the automatic transcription of the front-end recognizer to evaluate the recognizer's performance. Standard Precision (PRC) and Recall (RCL) measures are used to evaluate the segmentation performance of the system. These measures are defined as follows:

$$PRC = \frac{\text{Number of Correctly found boundaries}}{\text{Number of Hypothesized Boundaries}} \tag{2.5}$$

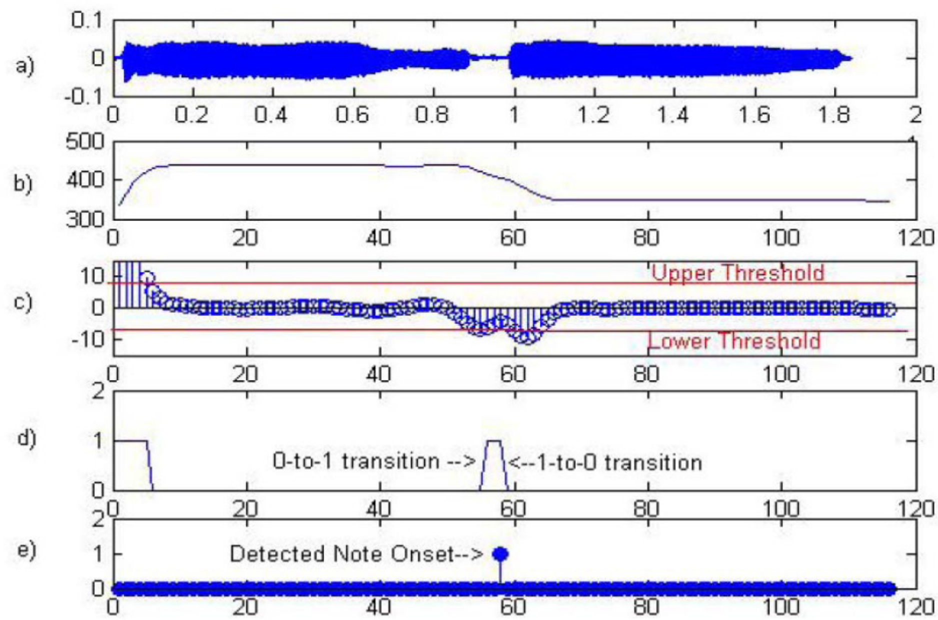


Figure 2.18: Example of deletion correction using pitch threshold filtering: a) two humming syllables which the recognizer considered to be one; b) corresponding pitch sequence P c) gradient vector, and threshold region $\tau - p$, d) Pitch change quantization;, and e)Onset detection

$$RCL = \frac{\text{Number of Correctly found boundaries}}{\text{Total Number of Boundaries}} \tag{2.6}$$

A time threshold value ΔT is also defined in order to define the location of a correct segmentation or boundary. A hypothesized boundary at time t is defined as correct if it lies within the time interval $t_0 - \Delta T \leq t \leq t_0 - \Delta T$, where is the correct boundary. In this study, ΔT is empirically chosen to be 75 ms, which is around 10% of the average note length in our test dataset (which corresponds to 3-4 frames of audio). 10% duration difference is assumed to be negligible in our experiments.

The F measure [vR79] is also calculated to provide a single performance metric over PRC and RCL.

$$F = \frac{2 \times PRC \times RCL}{PRC + RCL} \tag{2.7}$$

Segmentation tests were run for 10 rounds over 200 humming samples of “Happy Birthday”. In each round, 10% of the data was randomly selected, and the values of τ_p and τ_e (defined in Section 2.2.1 and Section 2.2.1) that maximizes the F measure were calculated. These values of τ_p and τ_e were then used in the tests of the remaining 90% of the data and an F measure for that particular round was calculated. After the 10 rounds were completed, results were averaged. For a 75ms tolerance window the performance of the recognizer and the improvement with the pitch and energy post processing is shown in Table 2.5.

Table 2.5: F measure results for segmentation performance

	F	1-F (Error)	Error Improvement
Speech Recognizer	0.79	0.21	-
Speech Rec.+ Pitch &Energy	0.84	0.16	23%

From the above table, one can see that on average 21% of the output sequence contained insertion and deletion errors. Post processing decreases the recognition error rate by a relative 23%. The recognizer with post processing achieves accuracy (F measure) of 84%. The performance improvement is confirmed to be statistically significant with p$\leq$0.01 using McNemar’s test for proportion comparison.

2.3 Search and Retrieval

The first task of the retrieval back end part of the system is to transcribe the segmented humming audio to musical symbols with pitch and duration information. The ultimate goal is to compare the transcribed audio to the database entries, and to find the closest melodic matches. Because of the afore-mentioned challenges such as user-dependent uncertainty and system-dependent segmentation errors, the query sequence may contain high levels of uncertainty. Such uncertainty affects the query calculations and can result in matching errors. For this reason, a robust retrieval system should be designed that can handle and accommodate such inherent uncertainty in the query.

2.3.1 Transcription

Conventional music notation provides a comprehensive way to represent a melody. The goal in our studies is to utilize an approach that is both comprehensive, and interconnects the query formulation in the front end with the database entries in the back end. As indicated in Figure 1.1, both the audio input and the database entries have to be represented in a consistent way so that meaningful symbol-to-symbol comparisons can be applied for search and retrieval. Inspired by QBH studies of other researchers [GLCS95, MSW^{+}96, MSWH00, RRG99, BR98, CJ98], relative pitch and duration information have been selected as the main attributes of a musical note in retrieval calculations.

In traditional music notation, absolute pitches are given names such as “A B C D” or are more commonly known (in absolute solfege) as “la, ti, do, re.” Such absolute notation is not helpful in a user centric domain, since few people have perfect pitch, and humming can be performed in any key. That is to say, the same melody can appear transposed any number of steps up or down based on the hummer’s vocal range and mood that day. Regardless, we can still recognize the original melody. Other researchers have validated the importance of relative pitch and duration contours in music perception [Ros02, Cap].

Consider “London Bridge is Falling Down,” shown in Figure 2.6. The original piece starts with a “re” (“D”) and continues with “mi re do ti do re” (“E D C B C D”). The original note sequence follows a semitone transition path of (+2 -2 -2 -1 +1 +2), as shown by the integer sequence on the same figure. The exact same melody can still be recognizably reproduced (sung or hummed) starting with any pitch, as long as the semitone differentials are performed reasonably accurately. Errors in note transitions decrease the perceived quality of the humming, and can lead to the intended piece sounding like some other melody. The desire to handle such variability in the human humming is one of the reasons that motivated our statistical approach to this problem, which will be detailed in Section 2.3.2.

After extracting duration and pitch values for each hummed note, we next label the note transitions. The Relative Pitch Difference (RPD) information, in semitones, is calculated using the formula:

$$RPD(k) = \frac{log f_k - log f_{k-1}}{log \sqrt[12]{2}}, \tag{2.8}$$

where f_k is the calculated frequency value of the k-th hummed note. Similarly, the Relative Duration Difference (RDD) is calculated as the ratio of the durations of two consecutive notes, and can be represented as:

$$RDD(k) = \frac{t_k}{t_{k-1}}, \tag{2.9}$$

where t_k is the duration of the k-th hummed note.

In contrast to common practice in the literature on melodic transcription techniques for QBH, we allow fractional number ratios because it accounts for finer levels of detail for user-centric variability, our main challenge in building a robust system. A human cannot be expected to hum a note transition perfectly, meaning that the logarithmic pitch difference will not necessarily correspond to integer ratios (cf. a “push-button” musical instrument such as a piano). This is one reason why we wish to record the variability as an unconstrained measure of error that is encountered during the humming.

The resulting transcription is a $2 \times N$ sequence (where N is the number of note transitions) with pitch transition information in the first row, and the duration ratios in the second row. A sample transcription output can be seen in Table 2.6.

Table 2.6: A section of transcribed output for a humming sample of “Happy Birthday”

RPD	0	1.70	2.44	-1.71	5.19	-0.81	-4.79	0.07	2.44	...
RDD	1	0.78	1.78	1.14	0.94	1.05	0.56	0.77	2.06	...

Without loss of generality, the first pitch transition is initialized to 0 and the first duration transition is initialized to 1, since the first hummed note has no prior reference. A positive pitch transcription corresponds to a pitch ascent from the previous hummed note and a duration ratio value that is larger than 1 refers to an increase in duration from the previous note value. The units for RPD is semitones, and there is no unit for RDD since it is ratio of two time measure.

2.3.2 Characteristic Fingerprints

In designing our search and retrieval algorithms collected data described in [UNC$^+$05] was analyzed. We performed a simple factor analysis of variance (ANOVA) for different control groups with respect to different criteria such as musical training, familiarity with the melody, and melodic complexity (perceived difficulty of melody). The statistical analysis of the collected data showed that the quality of the humming is dependent on people’s musical background and their familiarity with the melodies.

The analysis further showed that, regardless of musical training, more errors can be expected in the humming of larger note transitions than close intervals. It is also observed that humming major and minor intervals that occur more naturally and frequently in the scale, are easier than humming diminished or augmented intervals. Our plan is to incorporate these statistical findings in the design of our retrieval algorithms.

As mentioned in Section 1.1, different types of retrieval systems have been proposed including the popular DTW-ED methods proposed in [KNS$^+$99, ZS03, Pau02]. In this section, the performance of the DTW-ED algorithm will be compared to that of our proposed statistical retrieval approach. Distinct from our previous efforts on retrieval (see [UNC04]), the system reported in this work more directly integrates the statistical facts, gathered from detailed analysis of the humming database, into the search calculations. Specifically, our proposed approach to retrieval under the effects of production uncertainty is to collect fingerprints, i.e., the salient melodic information, from the transcribed symbol sequence, and to use them in searching through the database.

To identify the characteristic points in the hummed piece, aspects of music composition were considered. It is hypothesized that note transitions in a tune can be considered as distinct if they are novel (i.e., not frequent). For each humming query, the most distinct note transitions are located by searching the highest level of jump over RPD and RDD. The located jump on RPD will be able to capture the most distinct pitch movement and the highest jump on the RDD will be able to capture the most distinct duration movement in the query input. The selected FP's should be capable of identifying these local characteristic points in the hummed input. Note that large pitch and durations transitions are prone to error, thus having high uncertainty, a fact that can be explained by Fitt's Law [Fit54].

Fitt's Law predicts the time required to travel from a starting position to a target point area in space, as a function of the distance to the target and the area of the target. This movement can be considered as the vertical movement of larynx along the cervical spine for controlling f0 changes [HHMS99]. Vertical movement of larynx for high level pitch transitions is larger than the movement required for smaller pitch transitions. Therefore, given a constant time, the time it takes to finish a humming note and start the next one, users tend to make more errors while producing large pitch transitions.

The four generic finger prints selected in this work are created as follows:

$$
\begin{aligned}
(i_1, k_1) &= max(RPD), && (2.10)\\
(i_2, k_2) &= min(RPD), && (2.11)\\
(i_3, k_3) &= max(RDD), && (2.12)\\
(i_4, k_4) &= min(RDD) && (2.13)
\end{aligned}
$$

where k is the position of the critical element spotted in the query input and the i is the value of the corresponding element. For each k (k_1 to k_4) we create corresponding finger prints (FP_1 to FP_4) as follows:

$$
FP_j = \begin{pmatrix} RPD_{(k_j-R:k_j+R)} \\ RDD_{(k_j-R:k_j+R)} \end{pmatrix} \tag{2.14}
$$

FP's are selected sections of the transcribed humming sequence. Each FP is of radius R, where R is the number of note transitions in the transcribed humming sequence around the critical element k_i to be included as part of the FP. Hence the FP results in a matrix of dimension $2 \times (2R + 1)$. For example, if the largest pitch transition is located at k_1'th element of the input sequence, then the FP is a segment of the original input that spans the note transitions [$k_1 - R$, $k_1 + R$]. The first row of this substring carries the pitch transcription, RPD, and the second row the duration transcription, RDD.

The R value directly affects the length of the subsequence of the query that is assumed to be salient enough to predict the similarity between the selected Finger Print and its occurences in the database elements. To investigate the effect of R paramater, a target subsequence of a random input query subsequence is found in the database manually and co-occurences of its subsequences with different lengths has been counted from the 1000 melody database that is used in our retrieval experiments. The number of occurences will show how well the selected subsequence will represent the melody uniquely. Table 2.7 the mentioned humming input subsequence, and its corresponding target subsequences with different lengths. Fig. 2.19 shows the coverage graph for 3 information vectors of RPD, RDD and RPD+RDD.

Table 2.7: Random Subsequence and its corresponding rounded sequence

RPD	-4.62	0.07	2.44	-1.34	-1.11
RDD	0.56	0.77	2.06	1.5	0.51

Random input query subsequence

RPD	5	0	2	-1	-1
RDD	0.5	1	2	1.5	0.5

Corresponding target subsequence

RPD	2
RDD	2

A subsequence with length 1

RPD	2	-1
RDD	2	1.5

A subsequence with length 2

RPD	0	2	-1
RDD	1	2	1.5

A subsequence with length 3

RPD	0	2	-1	-1
RDD	1	2	1.5	0.5

A subsequence with length 4

RPD	-5	0	2	-1	-1
RDD	0.5	1	2	1.5	0.5

A subsequence with length 5 : the target subsequence itself

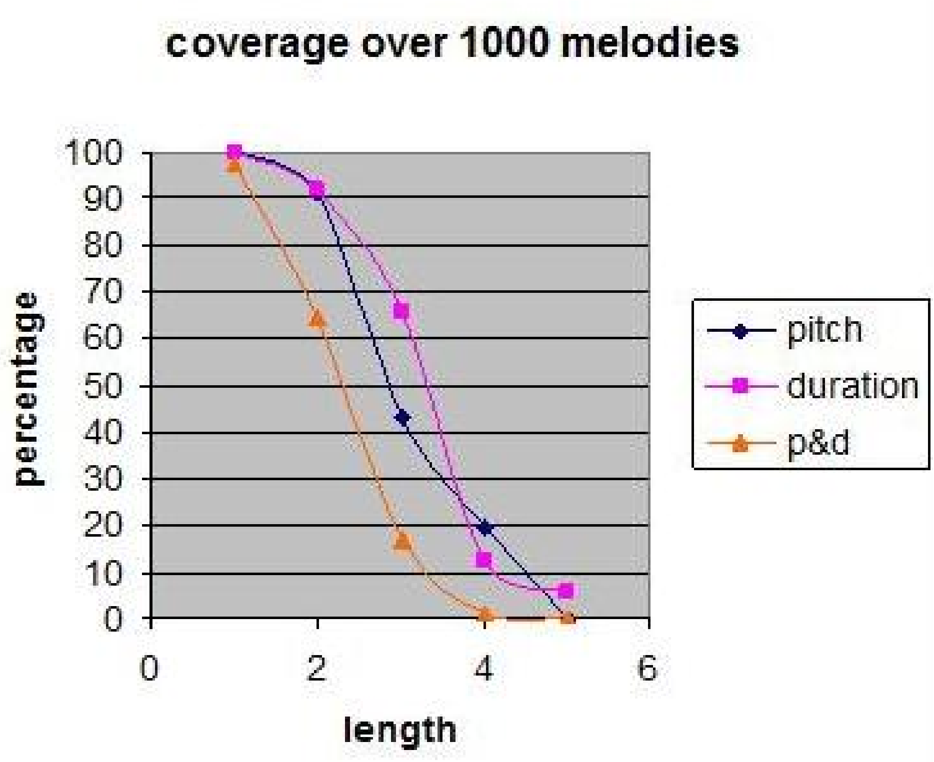


Figure 2.19: Counts of occurences of the different length subsequences in 1000 melodies database

The number of FP's selected in this work is four (FP_1 to FP_4), however, this amount can be extended for different characteristic points on the melody. While adding more FP's to the retrieval calculations can be helpful in finding better matches by introducing new characteristic sections of the query, the computation time increases dramatically since each FP is separately aligned to the database entries. Having less number of FP's may be computationally cheap however, the ability to carry distinctive melodic information with less number of FP's becomes limited. On average four FP's span around 10-15 note transitions since most of the time they overlap each other and was found to be adequate in our experiment.

In the employed FP's, minimum changes in RPD and RDD as well as the maximum changes are also considered. This is to account for the increasing importance and descriptiveness of duration ratio information as pitch variability is minimized, and similarly increased pitch descriptiveness when minimum duration variability occurs.

Table 2.8 shows some sample FP's extracted from a hummed sample of "Happy Birthday," of which a partial transcription is presented in Table 2.6. FP_1 is centered on the absolute highest pitch transition in the humming sequence, and has a radius $R = 2$, creating a 2×5 matrix using the neighboring two cells. FP_2 is centered on the smallest pitch transition, FP_3 is based on the largest duration change, and FP_4 on the smallest duration change. In earlier studies [UNC04], we have investigated the effect of selecting different R values. When R is too small, the FP does not carry enough information for the query calculations to be effective. On the other hand, when R is too large, the FP includes more uncertainty. Retrieval analysis showed that $R = 2$ is the optimal value for best performance.

Table 2.8: Sample Fingerprints (FP's)

RPD	-4.62	-0.33	**10.99**	-2.21	-3.71
RDD	0.48	0.86	2.42	0.94	1.06

FP1: largest pitch transition

RPD	-0.81	-4.79	**0.07**	2.44	-1.92
RDD	1.05	0.56	0.77	2.86	1.18

FP2: smallest pitch transition

RPD	-4.79	0.07	2.44	-1.92	6.54
RDD	0.56	0.77	**2.86**	1.18	0.91

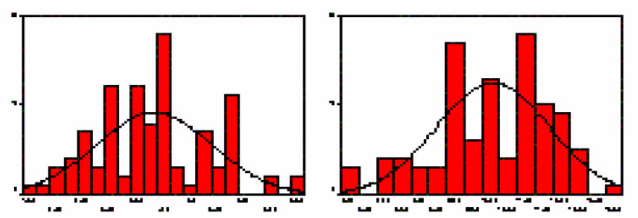
FP3: largest duration transition

RPD	-1.92	6.54	-1.82	-4.62	-0.35
RDD	1.18	0.91	**1.01**	0.48	0.86

FP4: smallest duration transition

2.3.3 Prediction Intervals

Considering the wide-ranging musical backgrounds amongst the data collection experiment participants, high levels of uncertainty and variability in the humming samples was observed. There exists a tradeoff between accepting high levels of variability, and achieving high accuracy in the retrieval system. High uncertainty may lead to radical changes in the interpreted melodic structure that could eventually result in erroneous matches in search calculations. Our solution to the handling of this tradeoff is statistical, and data driven. For each level of pitch and duration transitions, we build prediction confidence intervals (PCI) that are used in guiding the search engine as shown in Figure 2.20.



Transition	4 semitones	12 semitones
Mean	3.80	10.0
Std	1.08	1.80
PCI	$1.603 < x < 6.042$	$8.316 < x < 12.766$

Figure 2.20: Histogram and statistics for two note transitions(one 4-semitone transition, and one 12-semitones transition) each performed 100 times. For the bar graphs, horizontal axes are the semitones and the vertical axes are the number of occurrences for the corresponding semitone window within 100 performances.

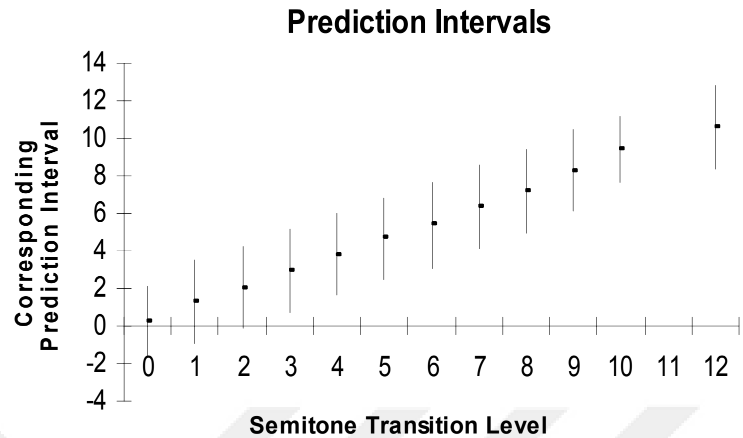
The note transitions performed by the participants, ranging from 1 to 12 semitones, are each found to be normally distributed with $p \leq 0.05$ according to the K-S test. We used 100 sample transitions, randomly chosen from our humming database, for training these prediction intervals. Figure 2.20 shows the prediction intervals for the cases of

4-semitone and 12-semitone pitch transitions in the humming of the melodies "London Bridge is Falling Down" and "Happy Birthday" respectively.

The prediction confidence interval for a 4-semitone leap (PCI[4]) gives the limits between which a performed transition may fall into the category of a semitone difference of 4. Referring back to FP1 in Table 2.8, since the center RPD value of 10.99 falls outside the allowable limits for PCI[4], the query engine will not search for a similar pattern around a 4 semitone level pitch transcription value in the database.

Figure 2.21 shows the distribution of the minimum, maximum, and mean number of semitone steps allowable for each semitone level, and a table of the calculated confidence limits for each pitch transition. These numbers are obtained from real world humming data, and do reflect the larger uncertainty introduced with non-musically trained users. For example, the performance for a unison transition is distributed within the limits [-1.62,2.19]. During regular humming performance, the accuracy of humming a pitch transition is largely affected by the previous and upcoming notes and the rhythmic structure of the melody at that particular point. This is why instead of collecting task specific data for certain transitions, transition data within melodies at random locations were collected, which lead to high levels of variance.

The prediction confidence interval for a 11-semitone transition, corresponding to a major 7th, was not calculated due to lack of data. For this case, a virtual distribution is created by linear interpolation of the other distributions in order to detect the sample distribution empirically when needed during error calculations. According to this table, the center value for FP1 (i_1 at k_1'th transcribed interval in the query input, in Table 2.8), 10.99, may represent a 10-, 11-(most likely) or 12-semitone transition. Prediction Confidence Interval calculations and such parameter estimations are done in a randomly selected development set and this development set was not used for performance evaluation tests.



Semitones	Number of Samples	Lower Confidence Limit	Upper Confidence Limit
0	100	-1.6152	2.1290
1	100	-.9192	3.5257
2	100	-.1258	4.2316
3	100	.7628	5.2031
4	100	1.6034	6.0422
5	100	2.4437	6.8817
6	18	3.0130	7.6954
7	100	4.1233	8.5615
8	100	4.9626	9.4019
9	100	6.1208	10.4510
10	24	7.6749	11.2312
11	-	-	-
12	100	8.3168	12.7666

Figure 2.21: Prediction Confidence Intervals. Sparse data for 11 semitone transition. Limits are in semitone units.

2.3.4 Error Calculations and Similarity Measurement

After all four finger prints are extracted, an error value for each of the finger prints against possible matches in the database is calculated. The procedure is demonstrated continuing with the FP_1 from Table 2.8. Having determined that the center pitch transition value of 10.99 (i_1 at k_1'th transcribed interval in the query input) could represent 10, 11, or 12 semitone leaps, the query engine will search through the database for these pitch transition values, $\{10, 11$ and $12\}$. For each candidate RPD value that is located, C is created, of size $2 \times (2R+1)$ in the same way FP's are extracted from the input sequence both considering the relative pitch difference RPD and the relative duration difference RDD. Next, the numerical difference, the total error, TE_j is assessed, between the query FP_j and the candidate matrix C_j.

In an earlier study [UNC04], the sum of the square differences between the components of the query FP and the original candidate matrix was used, C:

$$TE = \sum_{u=1}^{2} \sum_{v=1}^{2R+1} (FP_{uv} - C_{uv})^2 . \tag{2.15}$$

This previous approach does not take into account the difficulty in humming large pitch and duration transitions with high accuracy, as was found in our analyses of humming samples [UNC+05]. To account for typical errors made in the pitch transitions, rather than comparing the query with the exact pitch transitions, the FP's are matched against the expected production matrix, CM, instead of the original, C:

$$\begin{aligned} TE \ &= \alpha_1 \times \textstyle\sum_{v=1}^{2R+1} \left(FP_{(1v)} - CM_{(1v)}\right)^2 + \\ &\alpha_2 \times \textstyle\sum_{v=1}^{2R+1} \left(FP_{(2v)} - CM_{(2v)}\right)^2 \end{aligned} \tag{2.16}$$

where the first term calculates the pitch error and the second term calculates the duration error. α_1 and α_2 can be used in order to apply any weighting of preference. In our experiments the weights are selected to be equal. CM is given by mapping each exact transition to the expected value of its prediction confidence interval derived from actual humming data. No modifications are made for the duration data. The error value TE_j for each finger print FP_j is calculated.

Consider FP_1 from Table 2.8. Table 2.9(a) shows the most similar original candidate matrix, C, to the query "Happy Birthday", and Table 2.9(b) the corresponding expected production matrix, CM. The error value between FP_1 and the closest CM shown is $TE = 1.4391$.

Table 2.9: Candidate Symbol Sequences: C vs. CM

RPD	-5	0	12	-3	-4
RDD	0.25	1	2	1	1

(a) exact candidate matrix, C, most similar to FP_1.

RPD	-4.66	0.25	10.53	-2.98	-3.85
RDD	0.25	1	2	1	1

(b) expected production matrix, CM, closest to FP_1.

For each melody, q, in the database, if it contains the local feature within the defined prediction interval, a candidate segment $C(q)$ is extracted and mapped to its expected production matrix $CM(q)$.

An error value $TE(q)$ is then calculated between the query FP and $CM(q)$. Next, the candidate melody is assigned a local similarity measure SL by normalizing the TE values to "1" and subtracting the normalized value from 1 as follows:

$$SL(FP_1) = 1 - \text{norm}\,(TE) \tag{2.17}$$

This ensures that melodies with lower TE error values are assigned a higher local similarity measure SL, and melodies with higher TE error values are assigned lower SL values. Similar steps are applied to the other three remaining FP's for the same query (FP2, FP3 and FP4), and a composite similarity measure, S, is computed as the product of all four SL values for each candidate melody, q.

$$S(q) = \prod_{j=1}^{4} SL(FP_j) \tag{2.18}$$

In our system, the top f scoring melodies (we chose f=5 in our analysis below) are returned to the user as the query result.

2.3.5 Dynamic Time Warping, Edit Distance Approach

A common way to define a distance metric between two symbol sequences is the Edit Distance (ED). Several researchers, such as [Pau02], have adopted the ED method in their QBH systems for assessing similarity between transcribed audio and database entries. Since the input query and database elements are concurrent pitch and duration information, Dynamic Time Warping can then be employed to find the least cost alignment between the two, and the cost can be incorporated into the difference measure.

The ED between the compared sequences can be in the form of insertions, deletions, and transformations. With appropriate selection of the cost function, the ED can also represent the uncertainty that is expected in the humming audio input which takes user errors into account. Insertion cost covers extra hummed notes, while the deletion cost account for skipped notes. The transformation cost penalizes error between a performed transition and the expected reference transition.

Here, a DTW-ED algorithm is presented that is based on the earlier efforts in the literature, in order to make performance comparison analysis. The existing algorithms are modified and a version which is suitable to the way audio-to-symbol transcription is performed and to the way the melodies in the database are stored in our work was created.

This algorithm can be implemented by completing an $(M+1)\times(N+1)$ distance matrix $D(I,C)$ that calculates the distance between the $2\times M$ transcribed audio input I and the $2\times N$ candidate database entry CM . The following recursive formula is used to calculate the value in each cell in D:

$$
\begin{aligned}
i &= m+1 \\
j &= n+1 \\
m &= 1,2,3,...,M \\
n &= 1,2,3,...,N \\
D(i,j) &= \min\{D(i-1,j)+1, D(i,j-1)+1, \\
&\qquad D(i-1,j-1)+Cost(i,j)\} \\
Cost(i,j) &= \frac{1}{2}\left\{\left|\frac{I(1,m)-C(1,n)}{12}\right|\right\} \\
&+\frac{1}{2}\left\{\left|1-\frac{\min\{I(2,m),C(2,n)\}}{\max\{I(2,m),C(2,n)\}}\right|\right\}
\end{aligned}
$$

with initial conditions:

$$
\begin{aligned}
D(1,1) &= 0, \\
D(i=2:M+1,1) &= 1,2,3,...,M \\
D(1,j=1:N+1) &= 1,2,3,...,N
\end{aligned}
$$

The recursive formula above defines a constant penalty of 1 for note insertions and deletions, and a transposition cost for interval errors (normalized to one octave) and duration ratio difference cost. The $Cost$ function is the sum of the absolute pitch difference and the absolute duration ratio difference for the particular cell, normalizing to 1 to be consistent with the insertion and deletion penalty. The final ED between two sequences is given by $D(M+1, N+1)$.

2.3.6 Graphical User Interface

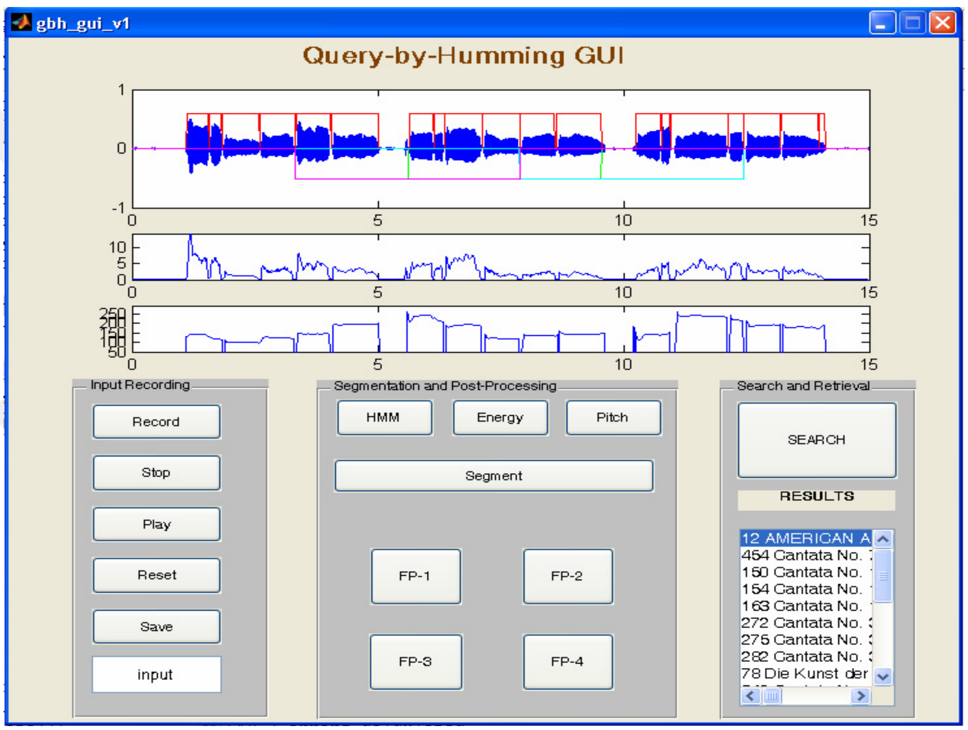


Figure 2.22: Graphical User Interface for the Proposed QBH System.

Figure 2.22 shows the Matlab Graphical User Interface (GUI) screenshot designed for the proposed system. The user can record his/her humming through the microphone, or audio samples can be loaded from a file. When the recording is finished or the upload complete, the $STOP$ button prompts the GUI panel to display the waveform, the short term energy and the pitch vector. This input can be played by the $PLAY$ button, or saved to a ".wav" file of name specified by the text block by pressing the $SAVE$ button.

The HMM, $ENERGY$ and $PITCH$ buttons perform the HMM-based segmentation and post processing. The $SEGMENT$ button draws the segmentation results on the input waveform. FP's can be extracted by selecting from the $FP1$, $FP2$, $FP3$, $FP4$ buttons; the system displays the corresponding FP package at the bottom of the waveform, and plays that segment of the audio input. The $SEARCH$ button performs the necessary error calculations, probability assignments and similarity assessment, and displays the top f results on the $RESULTS$ panel.

2.4 Experiments and Results

Both the FP and the DTW-ED algorithms are evaluated. As mentioned earlier, our evaluation corpus consisted of 400 samples selected from our humming database, distributed evenly between musically trained and non-trained subjects. These represent the melodies “Happy Birthday”, “London Bridge is Falling Down”, “Take Me Out to the Ball Game” and “Hey Jude”, 100 times each. Each humming sample is around 15 to 20 seconds long and does not necessarily start from the beginning of the main tune of the melody. Each target melody is in the database with 1500 more transcribed midi files.

Two different success measures were used: (A) the first considers a retrieval successful only if the correct melody is returned at the top of the result list; and (B) the second measure considers a retrieval successful if the correct melody is in the top f candidates. For A, the percentage accuracy shows: within 400 melodies what percent returned the correct song at the “top” of the result list (f=1) and for B, the percentage accuracy shows, what percent of the humming samples achieved a correct hit within top 5 candidate results (f=5).

Table 2.10 shows the results for the two measures applied to both the FP and DTW-ED algorithms, for both trained and non-trained participants. Tables 2.10(a) and 2.10(b) investigates the system's sensitivity or robustness versus increasing reference database size (MIDI collection).

Table 2.10: Accuracy Comparison for FP Algorithm vs. DTW-ED Method. Evaluation Methods: A - Correct retrieval tops result list, B - Correct Retrieval within top 5.

Test Database Size	500		1000		1500	
	A	B	A	B	A	B
FP	91	92	90	92	88	90
DTW-ED	90	90	88	89	86	89

(a) Percentage accuracy retrieval results for queries by musically trained participants.

Test Database Size	500		1000		1500	
	A	B	A	B	A	B
FP	73	77	72	74	70	72
DTW-ED	72	74	67	69	62	63

(b) Percentage accuracy retrieval results for queries by non- musically trained participants.

As expected, retrieval accuracy results for humming queries by non-trained participants using the same test database are consistently lower than those for queries by musically trained subjects. This is because the queries formulated by the non-trained participants contain more uncertainty, or maybe more prone to transcription errors.

In general, the FP method achieves higher retrieval accuracy than the DTW-ED method. For trained participants' humming queries shown in Table 2.10a, both the FP and DTW-ED approaches performed comparably well. This means that both algorithms work well in the presence of system-based errors and low levels of user-based production errors and the difference between the performances are statistically insignificant.

In the case of non-trained participants' (noisy) queries shown in Table 2.10b, the FP approach clearly outperforms the DTW-ED method. McNemar's test confirms that the performance difference is statistically significant with $p \leq 0.01$.

The results demonstrate that the data-driven statistical alignment of input to database entries via FP's makes the system more immune to user-dependent uncertainty. The main weakness of the DTW algorithm is the introduction of more uncertainty by considering the entire input data stream. A more robust approach should perhaps treat the input audio as a combination of characteristic structures. Our work demonstrates promising results with the introduction of the candidate structures of our FP approach, and encourages further examination of robust characteristics of musical signals.

2.5 Conclusion

This section presented a statistical approach to the problem of retrieval in the presence of uncertainty for QBH systems. Our research was motivated by the way humans perceive and reproduce music. Knowledge-based methods incorporating human understanding of music were combined with data-based models.

Our statistical approach required the extraction of FPs from hummed melodies. The search for these FPs was informed by our previous findings on inter-person humming variability, calculated from a database of humming samples by both musically trained and untrained participants. Our results showed that the size of the test database and the subjects musical training are the main factors that determined the success of our approach. We also implemented an ED approach for retrieval in order to compare the performance of the proposed FP algorithm to a more conventional approach. In our comparison for different test database sizes, the FP approach consistently outperformed the ED approach, and the results showed that the FP algorithm is more robust to user-dependent uncertainty, especially for the more noisy humming queries formulated by nonmusically trained participants.

One important question that one can ask at this point is how the performance of the retrieval system is affected by the accuracy of the front-end recognizer. For instance, for queries formulated by musically trained participants, where we can expect lowlevels of user-based errors, the dominant factor that impedes system performance are likely the system-based transcription errors, such as note insertions and deletions.

Future work will include further improvements on the front-end segmentation and recognition processes in order to achieve even better retrieval results, especially addressing the issue of mixed hummed and/or sung retrieval queries. We intend to use existing databases [26], our own, and combinations of the two to evaluate such schemes. The development of a scalable front-end segmentation algorithm will be critical to allow for unrestricted human audio input.

Moreover, better statistical models for the retrieval side of the system can also be developed, such as by extracting repeating patterns that could represent the most likely expected patterns observed in the users humming input [34]. This, in turn, implies higher penalties for unlikely database elements that will eventually result in improved retrieval calculations. Conversely, we could study the occurrence of rare melodic patterns as signatures to help enable fast retrieval.

Chapter 3

Query by Example

Due to advances in computer and network technologies, development of efficient data storage and retrieval techniques have received much attention in recent years. Music Information Retrieval (MIR) is one example of technologies that focus on identifying desired music data within large music collections. The query input to such systems may be of various types, such as modes of natural human interactions (humming, singing, recorded audio samples) or metadata (lyrics, genres, artists.) Given the metadata, retrieval can be straightforward; string matching algorithms that are used in web search engines are capable of these search and retrieval tasks. On the other hand, when the input query is in the form of audio, signal processing algorithms and music knowledge based techniques need to be incorporated. Query by Example (conver song identification) is the problem under discussion in this work, where a system must match a music query (a polyphonic signal) to similar music pieces or variations of the input query stored in a database. The variations might include different expressive performance parameters such as tempo, loudness, playing style or different instrumentations for the same music piece.

A considerable amount of research focused on the transcription of music audio signal to MIDI or piano roll type representations for accurate understanding of the tonal structure of a polyphonic tune. Numerous researchers modeled sound events in order to detect musical notes and their onset and offset times. Amongst them, Raphael [Rap02], Pertusa & Inesta [PI03], Smargadis & Brown [SB03], Ryynanen & Klapuri [RK05], and Poliner & Ellis [PE07] employed machine learning algorithms such as Hidden Markov

Models, Bayesian networks, and Support Vector Machines, which performed well for mono-timbral transcription tasks, such as piano music transcription, where only one single instrument is allowed. These results are promising, however, their extension to a general solution for non-instrument specific polyphonic transcription remains in question.

Since accurate transcription of multi channel audio is not easy, a mid level representation of audio is mostly desired for tracking tonal behaviour of the music pieces for query by example problem. Recent research attempts in [TYW02, Mar, PBC$^+$02, SG08, EU08] showed that, different representation techniques such as extracting dominant melody or a chord progression from the music piece could be a feasible solution for polyphonic representation, since, harmonic structure of such variations tend to be stable even the expressive parameters of the performance change. DTW, n-grams or edit distance type of string matching techniques were used for retrieval. On the other hand, [ECM08, SG08] used chorma based features to directly represent music pieces, without any musical labeling, while simple cross correlation of chroma vectors were used for measuring similarity. Kim also adopted delta chroma features in the representation which represents general movement in the harmonic structure of music pieces for more accurate representation and retrieval [SUN08].

The common solution to music audio matching and retrieval is to perform symbol-to-symbol comparison within a database to find the most similar, or the exact, matches to the input. Since the main features that are being used for the matching task are features (or symbols) that are extracted from the transcription process, the performance of the transcription directly impacts the performance of the matching and retrieval. In fact, retreival systems can be tolerant against some levels of uncertainty, so that the retrieval problem might be independent of the performance of accurate audio to symbol transcription.

Initial efforts in polyphonic music retrieval used MIDI transcriptions for modeling melodies. Doraisamy & Ruger [DR02] used MIDI transcriptions of musical pieces for comparing audio data; n-grams were built from different sets of features that were extracted from MIDI transcriptions, and the cosine rule was adopted for the ranked retrieval. Since MIDI is a text based representation of auido, where the note's exact locations (onsets and offsets), the timber and the intensitiy of the note is defined, the transcription from audio to symbols is not a challenge in here. However, the idea of statistical modeling of consecutive note events will be used in this work, in a similiar fashion, that will be explained more in detail in Section 3.2.

Pickens et. al. [PMS$^+$03] considered the query by example problem as a whole and proposed a general solution. They used existing polyphonic transcription systems in the literature to collect melodic note features from mono-timbral (piano only) music audio. The transcription was then mapped to a harmonic domain, namely a harmonic model that was designed for representing the auido with matrices corresponding to the distributions of the 24 lexical triads(three-note chords) and their transitions for the concurrent states. In later studies, Lavrenko & Pickens [LP03] used random fields to model polyphonic music pieces from MIDI files. Using random fields, they automatically induced new high level features from the melodies, such as consonant and dissonant chords, progressions and repetitions, to efficiently model polyphonic music information.

In this work, a similar strategy that involes a mid level tonal representation of audio and statistical tonal modeling for solving the query by example problem is used. The representation schema selected is slightly different from Pickens et. al.'s in that we prefer single dimensional representations of chord sequences, and the retrieval performance of the system is not directly limited by the performance of the initial audio-to-symbol transcription. Our aim for transcription from audio to meaningful symbols is a mid-level representation, which is independent of the exact note onsets and offsets, and also independent of the spectral effects of different musical instruments. Our optimization criterion in this work is not the transcription accuracy but the retrieval performance.

A frame level tonal representation is desired in this study. First, the audio is segmented into consecutive short windows, then, the short term frequency information is extracted. From the frequency vector, appropriate pitch information is recoreded, by searching for peaks in the frequency-power graph for each frame. The accumulated spectral information is used to construct a 12 dimensional feature vector which is the harmonic information extracted from the particular audio frame. For labeling each frame with a tonal name, the 12 dimensional spectral vector, namely the pitch class profile (PCP) is mapped on to the musical space (the Spiral Array) in order to find the exact location for the tonal center point in the 3 dimensional space. After calculating the tonal center, Chew's nearest negihbourhood algorithm is applied to identify the tonal name of the particular frame, by searching for nearest pre-defined Maj and min triad locations. After labeling of each frame in the music piece is completed, statistical n-grams are built from the text representation for each music piece. The database is a collection of n-gram models, each of which represents a single music piece. The retrieval is performed by simlply calculating the cross entrophy (the perplexity) of a text sequence, a transcribed audio query, with each of the melodic models in the database. From the resulting perplexity scores, an N-best of melodic models will be retrieved. The main goal in this study is to see how well this algorithm works for extracting an expressive performance of a meldoy from a large scale database. Fig. 3.1 gives a system overview.

The rest of the section is organized as follows. First in Section 3.1, the musical space (the Spiral Array) is discussed. The way the system uses the Spiral Array for creating the chord sequence for each music piece is explained in Section 3.2. The experimental setup is discussed in Section 3.3. Finally, in Section 3.4, the general discussion on the experimental results and future directions will conclude the section.

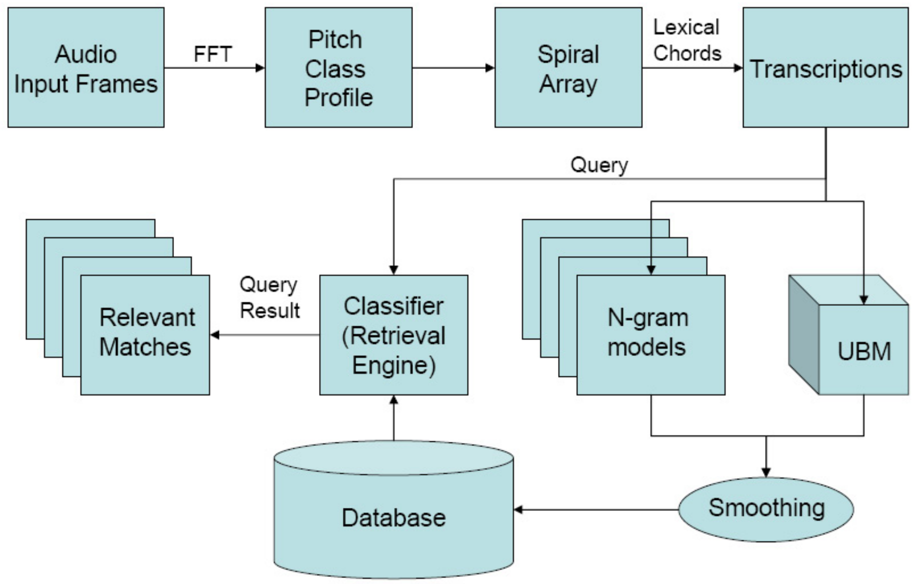


Figure 3.1: System Overview

3.1 Muscial Space: The Spiral Array

Through out history, mathematician and music theorists tried geometrically modeling the relationship between pitches and pitch related musical units such as notes chords and keys. The circle of fifths was the first credited model that represents the 12 notes in a circle with respect to their harmonic distance from each other using a perfect fifth ratio (3:2 pitch ratio). The perfect fifth ratio is assumed to be the closest harmonic distance between two pitches because of consonancy, that corresponds to the overlap between 2 sound signals. Circle of fifths is shown in Fig. 3.2

Later effort focused on defining multiple relations in one single model, and the tonal network (the tonnetz) was proposed by various researchers. The tonal network also includes Major 3rd and the minor 3rd harmonic distances in the tonal space and can be seen in Fig. 3.3. In the tonal network, each row corresponds to perfect fifth distances and each of them are placed in a position that the north east direction will be Major 3rd's

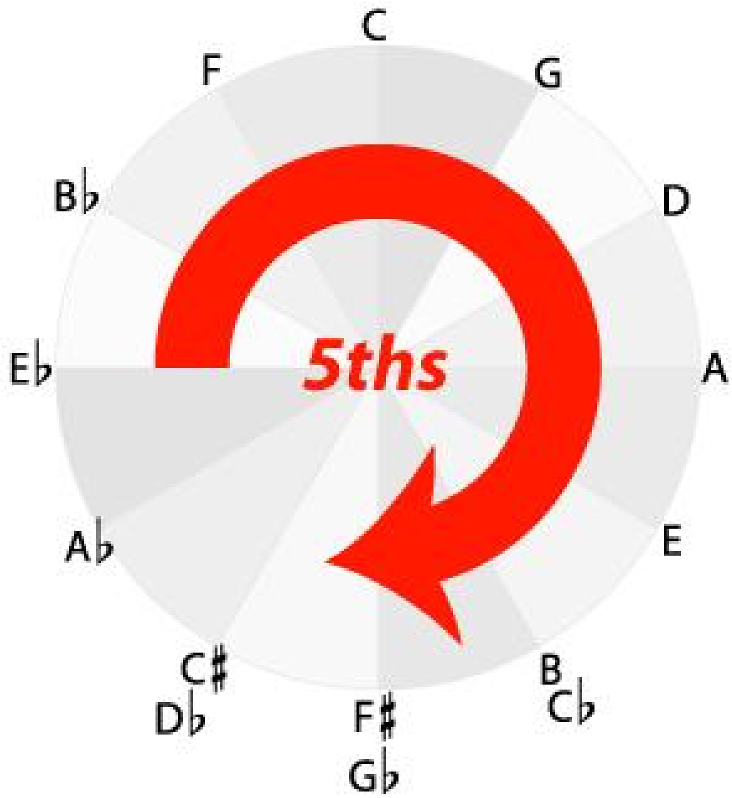


Figure 3.2: Circle of fifths: notes following perfect fifth harmonic distance in clock-wise direction

and the south-east direction will be the minor 3rd's. The raws of the tone network can be written continuously until it repeats itself from the same starting note which is a C. This repetation was first realized by Lubin, where he proposed concatinating the overlapping notes to form the tonal torus as shown in Fig. 3.4

In 2000, Chew proposed the Spiral Array, which places line of fifths (a single raw of the tonnetz) onto a spiral type of object, which has constant radius in xy plane and the degree of freedom on the z plane. The Spiral Array defines representations for pitches, chords, and keys in a three dimensional space. The Spiral Array has been used for key finding [Che01, CC05] and music similarity analysis [MC05]. We use the Spiral

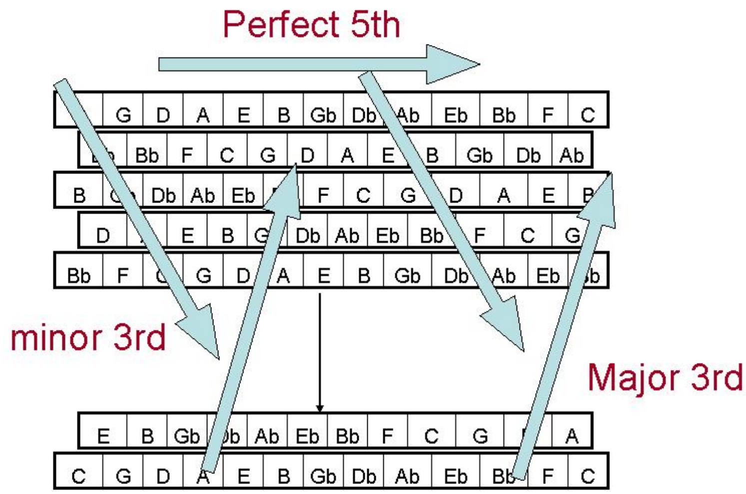


Figure 3.3: The Tonnetz (tonal/harmonic network)

Array to estimate musical chords (major and minor triads) for each frame set, and thus construct a one dimensional harmonic representation of the musical audio in the time domain. According to the model, as shown in Fig. 3.5, notes that are a Perfect 5th apart from each other are adjacent one to another on the spiral, and pitches that are a Major 3rd apart are vertical neighbors. Please see Chew [Che00] for other specifications.

On the spiral, 24 lexical chords are represented as triangles, an example of which is shown in Fig. 3.5. A Cmaj chord, consisting of C, the reference pitch, G, the perfect fifth, and E, the major 3rd for which, specific weights are assigned to generate a representative position for the particular triad, marked by a ”star” in the figure above. All such representation points for the 24 major and minor triads are computed inside the spiral.

The Spiral Array covers infinite number of notes that spreads on the z dimension in order to cover all possible pitch spellings in music theory for key identification and correct spelling purposes. Theoratically, different spellings of notes (for example, D^b

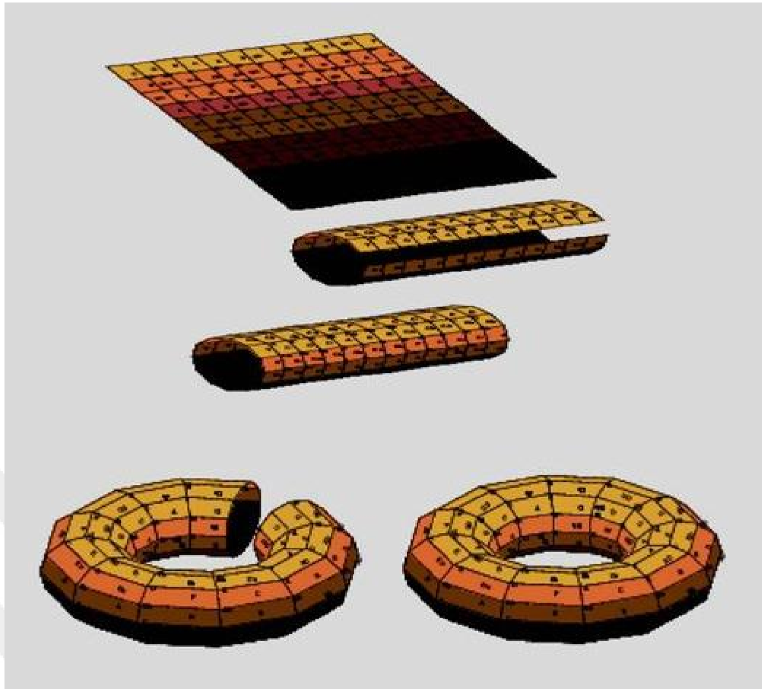


Figure 3.4: Construction of the tonal toroid from the harmonic network

vs $C^{\#}$) point different note names, however in practice, the physical pitch value for D^b and a $C^{\#}$ is exactly the same. This different choices in spelling becomes helpful in key identification problems [Che01], however, we do not seek to spell each note correctly in the music sequence, since only the absolute pitch values and the corresponding names are important in our harmonic representation. In order to eliminate repetitions of the spiral array in the z dimension, a packed version of this 3D tonal space is used. Our main interest is only over one full octave section of the Spiral Array, and this 12 note section is connected from top to bottom, in order to create the 3D toroid object musical space 3.6. From the fixed positions of the 12 notes on the spiral toroid, triad triangles can be constructed and their associated centroid (center of weight) points is calculated. The centroid for each triangle is located at the intersection of the medians of the triangle

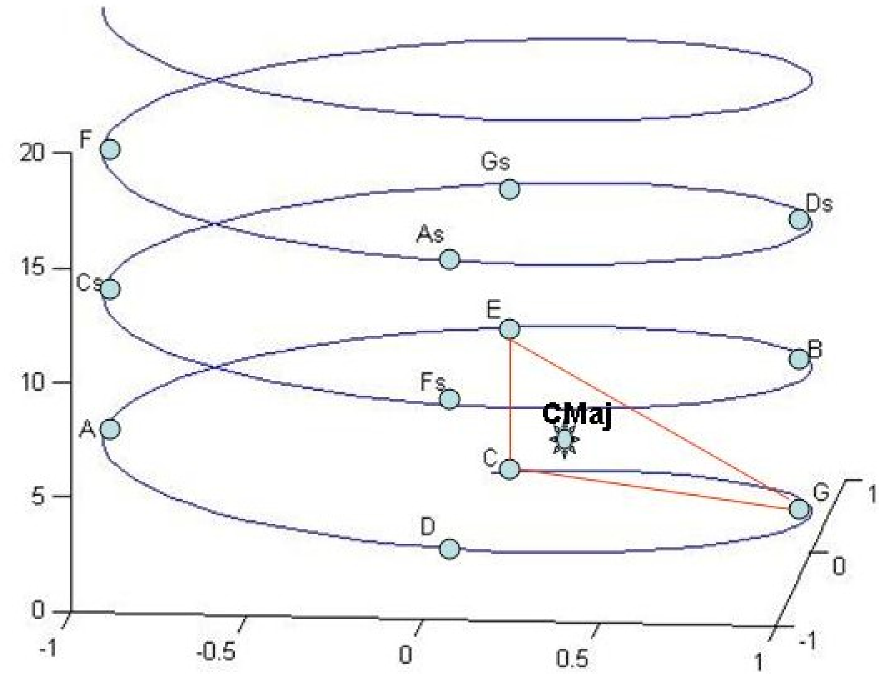


Figure 3.5: Spiral Array: pitch locations and the C-Major triad chord

edges and they are the representative location of that particular triad, that will be used later in distance calculations.

The geometry can be constucted with the following command line in matlab shown in 3.1, where rz is the radius of the inner circle of the torus, and the rx is the radius of the outer circle of the torus, and $rb = (rx + rz)/2$, which also makes $rb = rz + ry$, where ry is the radius of the circle that is visible at the cross section of the torus in z dimension shown in Fig. 3.7. The values of the radii are assigned to make sure, the distance between the adjacent notes which are harmonically perfect 5th distance away from each other are the minimum distance that can be defined on the torus, and this distance is followed by the Major 3rd distance and this is followed by the minor 3rd distance. This distance alignment also makes sure that, for each constructed triad, the centroid is the closest to the root note. In our experiments, we set the rx=100 and rz=80, which are the optimum numbers that are calculated in our validation experiments.

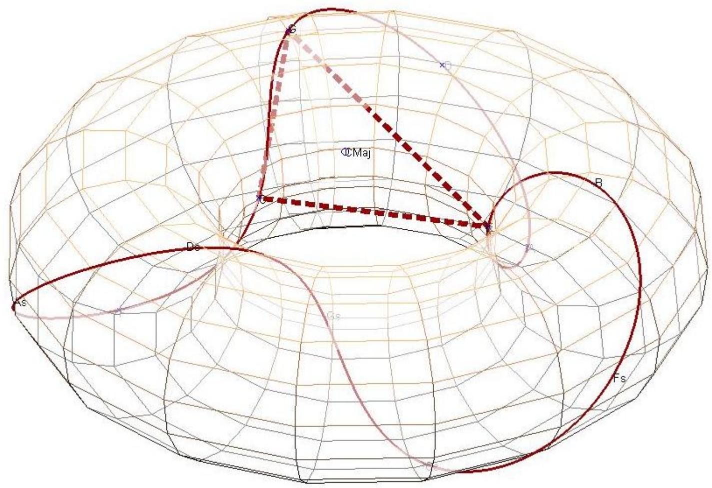


Figure 3.6: Toroid: 12 distinct note positions and construction of the CMaj triad with C, G, and E. Not all notes are visible.

$$
\begin{aligned}
[u,v] &= meshgrid(-pi : 0.4 : pi + 0.4, -pi : 0.4 : pi + 0.4); \\
tx &= (rb * ones(size(v)) + rz * cos(v)). * sin(u); \\
ty &= (rb * ones(size(v)) + rz * cos(v)). * cos(u); \\
tz &= rz * sin(v);
\end{aligned}
\tag{3.1}
$$

with the location of the notes are defined with following command lines in 3.2:

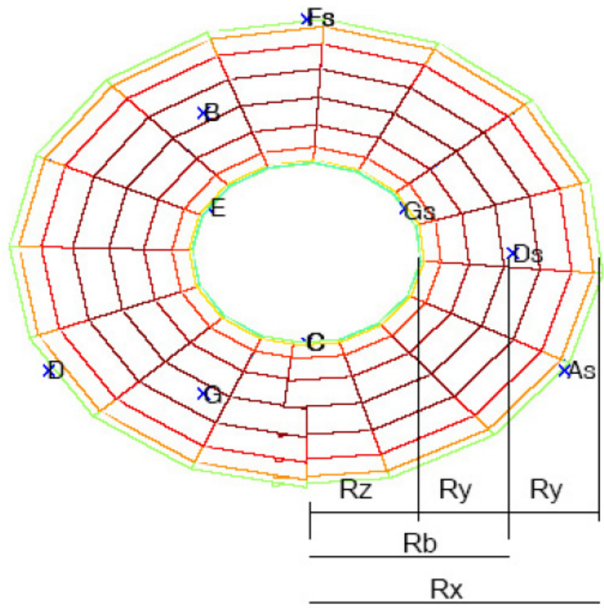


Figure 3.7: Defined radius values for the circles that creates the torus

$$
\begin{aligned}
c &= [rz * cos(pi * 3/2), rz * sin(pi * 3/2), 0]; \\
g &= [rb * cos(pi * 8/6), rb * sin(pi * 8/6), ry]; \\
d &= [rx * cos(pi * 7/6), rx * sin(pi * 7/6), 0]; \\
a &= [rb * cos(pi * 6/6), rb * sin(pi * 6/6), -ry]; \\
e &= [rz * cos(pi * 5/6), rz * sin(pi * 5/6), 0]; \\
b &= [rb * cos(pi * 4/6), rb * sin(pi * 4/6), ry]; \\
fs &= [rx * cos(pi * 3/6), rx * sin(pi * 3/6), 0]; \\
cs &= [rb * cos(pi * 2/6), rb * sin(pi * 2/6), -ry]; \\
gs &= [rz * cos(pi/6), rz * sin(pi/6), 0]; \\
ds &= [rb * cos(pi * 0/6), rb * sin(pi * 0/6), ry]; \\
as &= [rx * cos(-1 * pi * 1/6), rx * sin(-1 * pi * 1/6), 0]; \\
f &= [rb * cos(-2 * pi * 1/6), rb * sin(-2 * pi * 1/6), -ry]; \qquad (3.2)
\end{aligned}
$$

3.2 Melodic Representation

Because of the complex nature of polyphonic music audio, a direct mapping from audio to musical notes is not straightforward. As mentioned in Section 1.3, researchers have attempted to solve the polyphonic transcription problem using a variety of techniques, but their success has been primarily limited to mono-timbral experiments. For this reason, a mid-level representation is chosen that can be generalized for any kind of instrumental music audio. Similar to a previous approach by Pickens et. al., 24 lexical chords is selected as the representation grammar, spanning all major and minor triads in one full octave. As shown in Fig. 3.1, the audio pieces are first segmented into small frames, then each frame is stamped with one of the 24 lexical chords. For chord estimation, frequency spectrum of the corresponding frame is mapped onto the Spiral Array representation, the details of which will be addressed in the next section.

3.2.1 Feature Extraction

62.5ms non-overlapping hamming audio windows are used and fft algorithm is applied to gather the frequency spectrum of each consecutive frame. Fig. 3.8 shows the fft spectrum of a random frame in the Molto Allegro movement of Mozart's *Symphony No.40*.

In the frequency spectrum shown, some peaks are marked with note information on the active pitch (not all are annotated). These peaks are automatically selected by applying a simple curve fitting algorithm on the fft vector. A pitch range from 27.5 Hz (A0) to 3520 Hz (A7) is considered. The system uses a short length sliding window

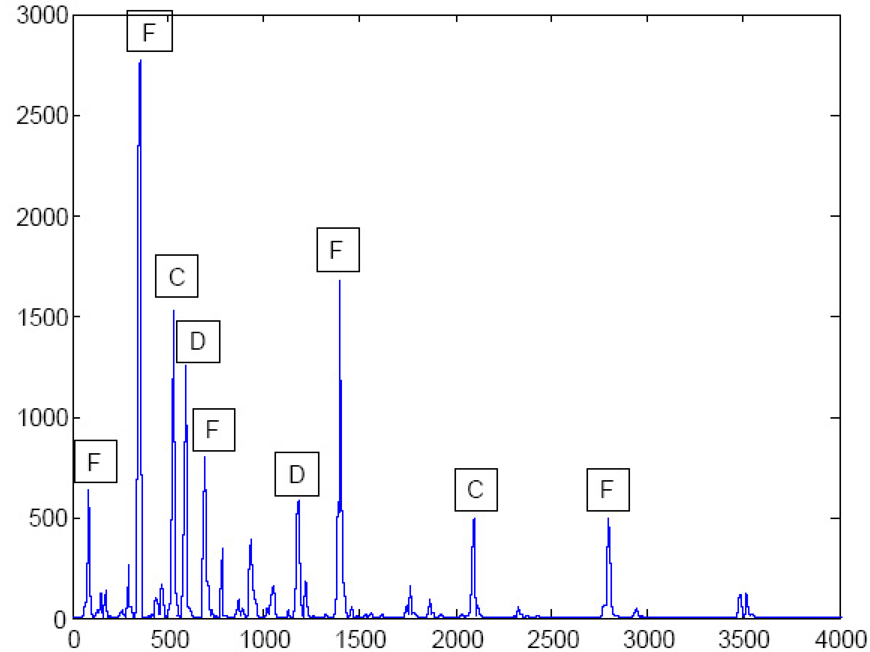


Figure 3.8: Frequency spectrum of a random frame generated by fft with note annotations based on peak detection

for the fft vector. For the samples inside the window, we fit a parabolic function to the sample points. If the parabolic function is concave and the maximum (the point at which $f'(x) = 0$) lies within the selected fft window, then a peak is assumed to exist. The active pitch in this location is the one that corresponds to the maximum point of the windowed fft vector.

For each detected peak, its active pitch and its amplitude is recorded. After all the possible peaks in the frequency spectrum are extracted, the information is accumulated in a vector called the pitch class profile (PCP) which contains information on the energy and the notes detected. The corresponding PCP for the above frame (Fig. 3.8) is shown in Fig. 3.9.

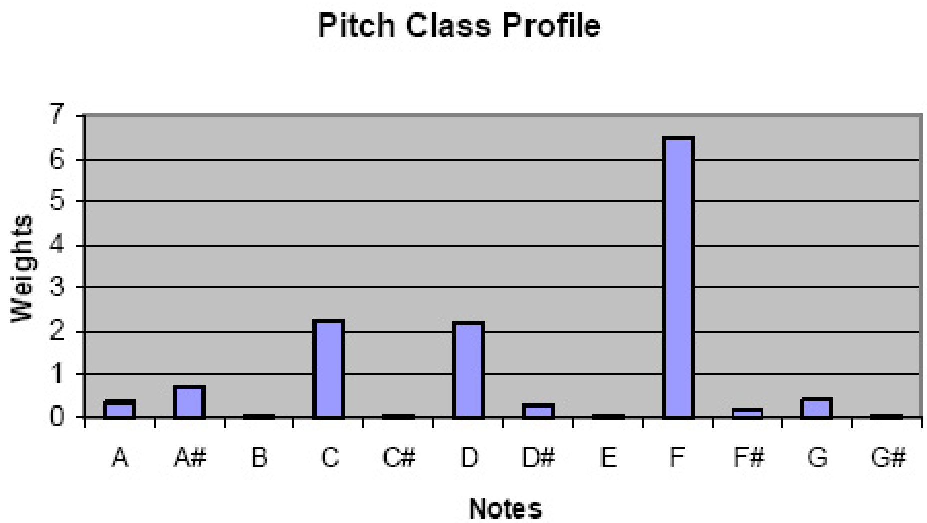


Figure 3.9: Corresponding PCP vector for the frequency spectrum of Fig. 3.8 that shows weights of 12 distinct pitches

3.2.2 PCP to Spiral Array

One can see from the figure that, F, C and D are the most dominant notes in this particular audio window. Now, given the weight profile of the active pitches, our goal is to assign the most meaningful triad to this particular frame, and with the remaining frames, to extract a one dimensional representation for the whole melody. The Spiral Array/Toroid model is used to achieve this goal. From the PCP vector, each calculated weight is mapped on the torus, and a center of weight is calculated in 3D space as shown in Fig. 3.10. The calculated center of weight represents the tonal center of the particular frame and our system aims capturing and recording the movement of the calculated consequent tonal centers, after they are named with the most appropriate triad. System choses the most closest triad centroid positions that are already calculated and fixed for labeling (nearest neighbour). This compansates the affect of different spectral characteristics of different instruments playing same harmonic movements. Also the intensity levels of the played instruments should not change the harmonic movement, since the weights of each note captured by the pcp should increase or decrease propotionally for passages played with different intensity, which eventually rests close to the main harmonic movement in 3D space.

The resulting representation is a n dimensional text representation of the audio pieces as shown in Fig. 3.11. In the figure, the first bar of Concierto en la menor is shown with the true transcription of the harmonic movement into triad representation in the first line under the score. The following two lines show, the representations gathered from two different expressive performances in our database, the first played with pianos with beat value is set to 120b/s, and the second played with a small orchestra consisting of 2 violins, one organ and one harpsicord with beat value is set to 80b/s, respectively. From the figure, one can see that the version which is performed slower, produced more labels for each played sound. Note that, our system considers one sixteenth of a second as one frame. The transcription also includes insertion and deletion errors that is generated by the output of the system.

3.2.3 Validation Experiments

In order to validate the classification accuracy of the tonal space model, a number of experiments was conducted from synthesized midi data. A musical note sequence that includes different combination of notes in one octave was randomly generated and each note group was hand labeled with the corresponding triad level representation. This experiment was conducted through a midi synthesizer which is able to produce the same note sequence with different instruments. The note sequence was repeated multiple times with available instruments in the synthesizer (piano, guitar, violin, church organ, etc...). From the hand labeles we compared the performance of the frame level automatic transcription to the hand labels and used PRC and RCL to calculate the efficiency of the chord classification algorithm. Even though this experiment provides noiseless input

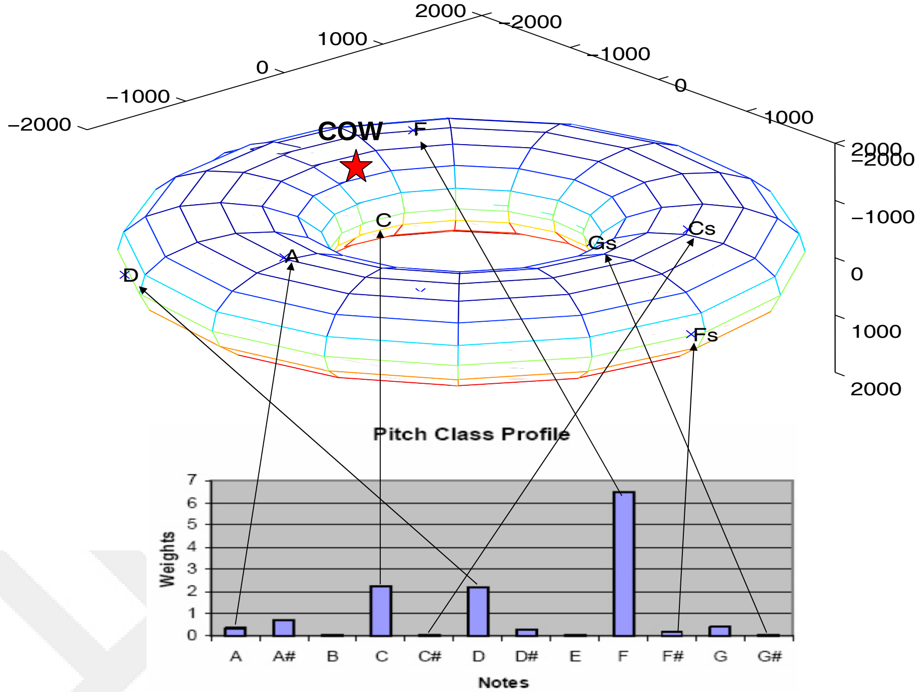


Figure 3.10: Calculation of the position of the Center of Weight in 3D space from the pcp vector

data to our classification algorithm, we believe that it is a reasonable way of validating the functionality of the system.

Under the assumption that a single note will rest to its Major 3rd chord and a 2 note chord will rest to a Major 3rd if the root and the 5th or the root and the major 3rd exists, the classification accuracy for monotimbral polyphony (one instrument multiple sounds) is around 92 %. For multitimbral polyphony (multiple instruments multiple sounds) the classification accuracy is 76 %.

These results suggest that we are able to detect the chord progression of a polyphonic audio signal with reasonable accuracy, and it is feasible to use this type of symbol level transcription for retrieval purposes.

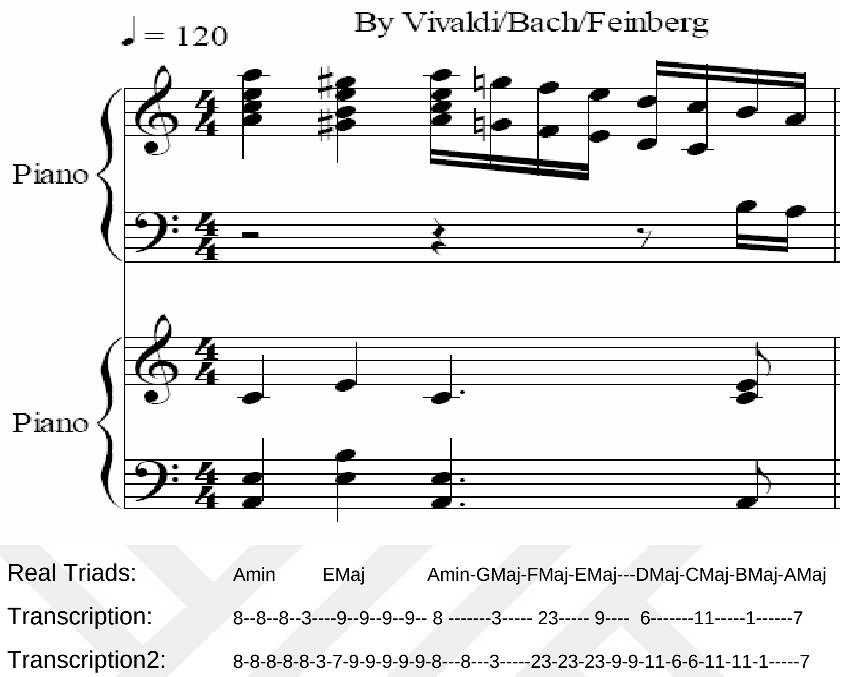


Figure 3.11: First bar of ”Concierto en la menor” With the real triads, and the frame by frame transcription of two expressive variations. Triads decoded with numbers 1 to 24.

3.2.4 Normalization

Since any music piece can be played in any key, the system needs to consider for normalizing musical key differences of the input query and the available variations in the dataset. In this system, a straightforward normalization technique is used by simply, transposing each representation to a fixed triad, which is in this case CMaj. This is done by, finding the most frequent harmonic triad in the representation (the mode of the representation vector) and tranpose it to the CMaj by shifting each label by the difference between the CMaj and the ”mode” of the text representation V_l, where l is the frame index of the query piece as can be seen in 3.3.

$$
\begin{aligned}
shift &= mode(V) - 1 \\
V &= mod((V - shift), 24) \\
V(V == 0) &= 24
\end{aligned} \qquad (3.3)
$$

3.2.5 Modeling

An n-gram is a statistical model of subsequences of n items within a larger sequence, and is in common use in natural language processing applications to model word sequence statistics. We use n-grams to statistically deduce information of the harmonic behavior of polyphonic music pieces. While a music piece is played slow, more frames with similiar harmonic labels will repeat itself, however, when the same melody is played faster the number of labels decrease. The sequential statistical models are assumed to compansate for this variable lengths in the representation.

As seen from Fig. 3.1, polyphonic pieces in our database is stored in the form of n-gram models in order to quantify the likelihood that a given query sequence of chords is generated by one of the stored melodic models.

To enable the efficient use of this strategy, normalization of the n-gram models is required. For normalization purposes, a Universal Background Model (UBM) is produced by concatenating all the available text transcriptions into one single document, and creating an n-gram for this particular collection. By mixing the UBM with each individual melodic n-gram model using a low weight, the required smoothing is performed. Finally, the collection of the smoothed melodic n-grams constitutes our database.

3.2.6 Perplexity-based Evaluation

Perplexity is a common way of evaluating the complexity of language models (i.e., its branching factor). In this work perplexity is used to evaluate similarity of the melodic models against a given query chord (triad) sequence. The perplexity measure gives the likelihood that the query was generated by a specific probability distribution, namely one of the melodic n-gram models. By the calculated perplexity scores, our retrieval engine gives an N-best list of most likely melody candidates. For creating the n-gram models, performning smoothing by the UBM, and model evaluation, we used the SRILM toolkit [Sto02].

3.3 Retrieval Experiments Setup

This section describes the evaluation of our system.

3.3.1 Data

6027 polyphonic music files are downloaded from classical-archives website in MIDI format. Around 2591 of these files include 937 original music pieces and their expressive variations (i.e Table 3.1), and the rest of the 6027 are music pieces without any expressive variation. Some pieces have only one variation while some others have up to 13 variations. The system will only be tested for the music pieces which has any variations available in our representative database. All MIDI files are converted to $wave$ files using $timidity$ and stored with 16khz sampling rate.

Table 3.1: Melody: Mozart's 40th Symphony, molto-allegro. Descriptions of the variations in our database

Variation	Description
v0	orchestral performance
v1	piano variation
v2	orchestral performance, different instruments and expressive style
v3	piano
v4	piano duet

3.3.2 Retrieval Tests

This section explains the tests that have been performed to validate the functionality of the algorithm, the validity of the representation technique and its usability for modeling polyphonic melodies. First, the plain text representation for a music piece is used as a query, and the output of the system is observed to see if any of the expressive variations are seen in the N-best list. For instance, when using Mozart's *Symphony No.40* as a test sample, we select "version 0" amongst all relevant documents in the database as the input query, and checked to see if the resulting score table contained versions $1, 2, 3, \ldots, m$ in the N-best list. Setting N to different numbers, the same strategy is applied to all variation groups in the database, and the results are reported in next subsection. Correct retrieval occurs when one of the target models is in the N-best list, otherwise the results is classified as an incorrect retrieval. Also, in information retrieval, in such evaluation systems, a more detailed way of testing the retrieval performance is testing for the overall rank of the correct match within the N-best list, which is called the Mean Reciprocal Rank (MRR).

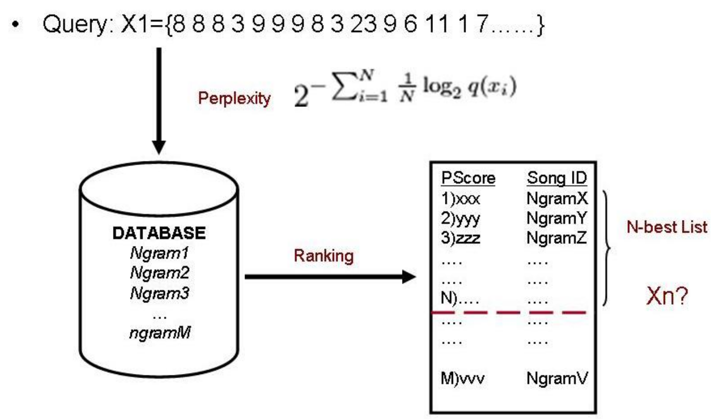


Figure 3.12: Test 1: Retrieval using the plain representaiton

Retrieval with plain representation

Fig. 3.12 shows the flowchart of the first set of tests where the plain representation is used for the retreival task. The plain representation is a 2 dimensional representation which includes information in time and tonality axises compoundly. Since we have a frame level representation, each frame corresponds to an instance of a tonal state and the continuity of that state suggest the length of that tonal decesion in the time domain which might suggest information about the rhythmic characteristics of the music piece. However, the resulting representation which is a mixture of time and tonal information is shown to have been not perfect by the validation experiments. The probability of an error to occure in noiseless input (synthesized midi converted to .wav auido) is around 26% and this error ratio is calculated for harmonic identity. The harmonic error directly affects the error probability in the rhythmic information because the repetitions are counted incorrectly due to insertions and deletions. Even though we have also tested using the rhythmic information for retrieval purposes, we concluded that the two dimensional information carries very little accurate rhytmic information thus can not be used for meaningfull rhythmic level comparison. We also concluded that, rhythm should be considered separately, aside from the harmonic representation and should be extracted using more robust features such as local energy spectra.

The transcribed frame level harmonic information can be seen in the Fig. 3.13 which consideres the transcription a compound information that is generated in the time axis and the harmonic axis.

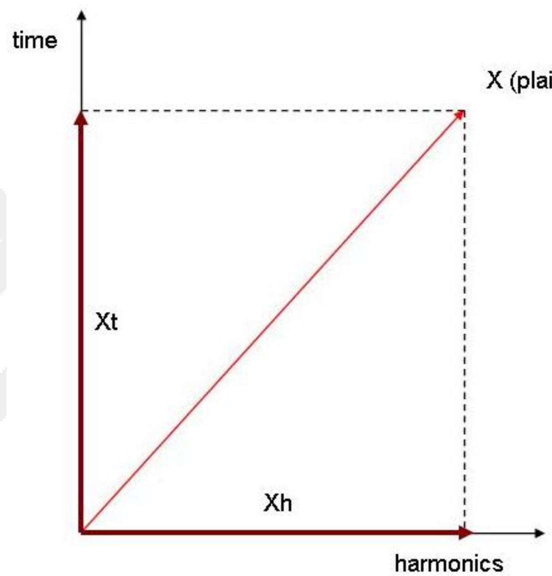


Figure 3.13: The frame level harmonic representation includes time and harmonic information at the same time

Retrieval using supporting harmonic models

For the second set of tests, we would like to see the effect of tempo variations by completely eliminating it from the text representations by deleting repeated triad labels. Since the N-grams statistically represent only 3 consequent frames (N=3), the tonal movement will be recorded for only 3 consecutive frames, for tonal labels that repeats itself for more than 3 times, the N-grams do not record any tonal movement. By eliminating the repitations, the representation is forced to cover transitions between harmonic states. This can be explained more clearly in Fig. 3.14. The experiment setup for the second set of tests can be seen in Fig. 3.15

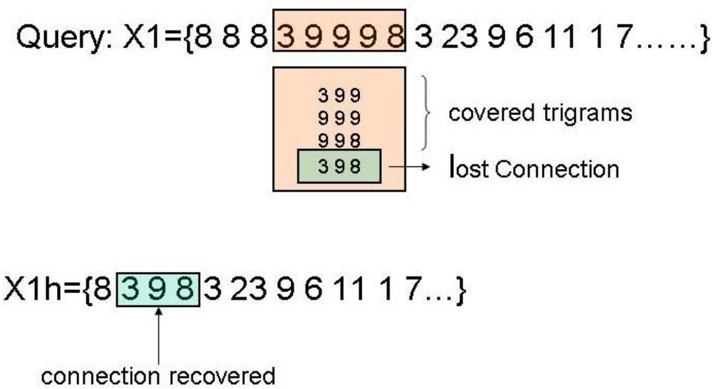


Figure 3.14: Repetition elimination from the representation. X1h represents the pure harmonic state model without any temporal information

Retrieval using random short clips of audio queries

The next set of tests are designed to see a more practical evaluation case, where the query to the system is a random short clip of the original melody which are 15s, 25s, and 35s. By varying the length of the clip, we also aim to see the effect of increasing input length over retrieval performance. For, this a random point in the original melody is selected at time t, and 3 clips that spans t, $t + 10s$ and $t + 15s$ are created. Also the target models are different this time, where generic models are created by mixing all the available N-gram models for each versions of the same melody with equal weight, leaving the input query out. For this particular test, there are 937 query pieces, and 937 associated generic models created by the available versions of the original piece, and around 3436 exra music pieces without any variations. In total there are 4373 music models in the database and the performance results are given for this particular dataset. Here the application will be slightly different, since thre will be only one generic model

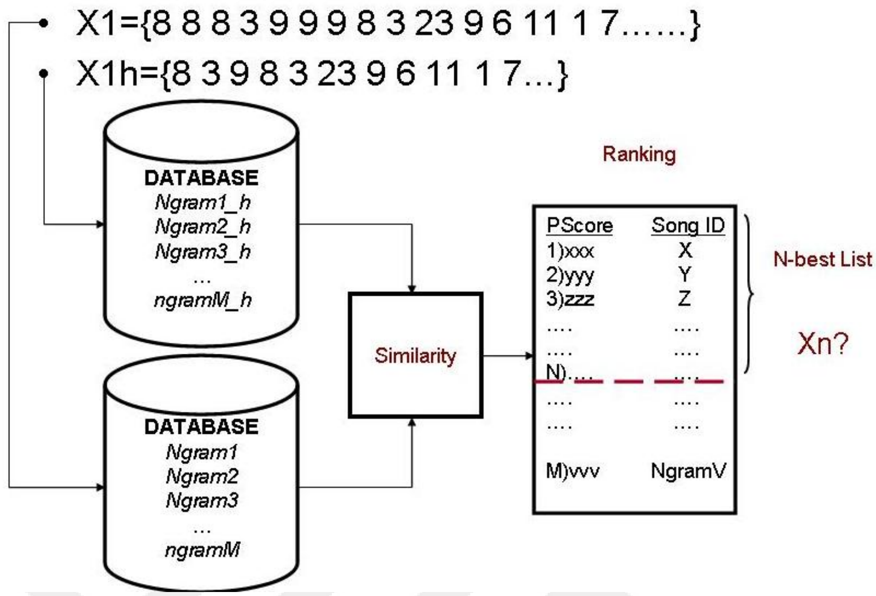


Figure 3.15: Test 2: Retrieval using the supporting harmonic models with the plain models.

for each melody as the query target. Instead of trying to retrieve relevant variations, we would then aim to match to a single model for each query input with short clips from the original melody. This approach could result in more meaningful retrieval in the face of extremely large databases.

Retrieval using real recordings as query

For the last experimental test, we have collected 100 real recordings from the web as mp3 files for 100 of the melodies that we have in our database corpus. These real recordings are performed in studios or concert halls, which makes the sound quality very high. We used this small real data corpus as input query to our system and we would like to see if we can gather relevant variations that were constructed by midi data. The difference between the real recordings and the midi synthesized audio is obvious, however, this testing strategy will tell us if our algorithm is robust against signal level variations.

Retrieval using supporting time models

In this set of experiments, we extracted time representations from the plain lexical chord representation and used it as supporting models to the final decision of the perplexity based classifier. Being encouraged by the positive affect of using supporting harmonic models, we expected the time models that are extracted from the plain lexical chord representation to help boost the performance of the retrieval system. However the experiments showed that, the time models that are directly extracted by counting the repeated harmonic states of the plain representation was not accurate enough to help the retrieval side of the system. We explained this fact by performing an error analysis in the next section 3.4.

3.4 Results and Discussion

Following are the result tables for each of the experiments mentioned above. As mentioned earlier, the retreival tests are applied for the music pieces which has a valid expressive performance version in the database, and each of these pieces are tested for retrieving each and every of its own versions. The results are reported with percentage values with respect to $accuracy$ and mrr.

As expected, from Table 3.2, the retrieval result improves when tolerance region in the N-best list is increased. It is critical to set N to a reasonable level, since in practical use, the number of samples in the database can be extremely large. Even for a small scale database, an 70.6% retrieval accuracy for $N = 60$ (1% of the database size) is promising, considering the different types of variations that are being tested. For the

Table 3.2: Retrieval Test Results for whole melody inputs where a correct match is defined when the N-best list includes the target polyphonic audio

	Length of the N-best List		
	N=10	N=60	N=100
Accuracy	57.8	70.6	73.7
MRR	61.2	65.8	68.4

second tests, we used the perplexity calculated from the first set of experiments whose results are reported in Table 3.2 and combined the perplexity values with the updated representation that is gathered by elimination tempo variations. The new models boosted the retreival performance as shown in Table 3.3

Table 3.3: Retrieval Test Results for whole melody inputs where a correct match is defined when the N-best list includes the target polyphonic audio

	Length of the N-best List		
	N=10	N=60	N=100
Accuracy	68.4	71.8	74.2
MRR	65.4	66.1	68.8

When one compares the results reported in Table 3.2 and the Table 3.3, it can be seen that there is a promising improvement, especially for N=10, which is a strict limit for the retriveal performace for a database size of 6027 music files. the 7.6% absolute improvement retrieval shows, the N-gram models created with the updated representation were able help the system, for improving retrieval performance with the cost of computation time.

For the test set that includes random short clips with 15s, 25s and 35s lengths from the pieces, the effect of different input lengths is examined. Results for $N = 60$ are reported in Table 3.4.

Table 3.4: Retrieval test results for short clips

	Length of the query		
	15s.	25s.	35s.
Accuracy	67.8	72.1	75.4
MRR	66.2	67.8	70.4

As can be seen from Table 3.4, for 25s clips, we achieved 72.1% retrieval accuracy within the top 60 of the results list. When the length of the input query increases, the retrieval accuracy improves as expected; this is because more data in the input sequence provides more meaningful comparisons in retrieval. This relative change may be more significant in the cases where the number of samples in the database is large.

For real recordings test case, we would like to see if we can capture at least one of the available variations in the top N best list first, and calculate the MRR value, and also we would like to see how much percent of the available target variations can we obtain in the N list with a calculated MRR value. Table 3.5 shows the experimental results for the real recordings test case.

Table 3.5: Retrieval test results for real recordings test set

	At least 1 variation	All available variations
Accuracy	84	53.3
MRR	86.2	61.8

As seen from Table 3.5, 84 of the 100 real recordings was able to capture at least one relevant variation in the 6027 song database, and from the total 754 target variations, we were able to get 53.3% of them in the top N=60 of the result list with MRR values, 86.2 and 61.8, respectively. Considering the difference between the real recordings and the available midi synthesized variations, the results are promising.

3.4.1 Error Analysis for time models

From our validation experiments explained in Section 3.2.3 we presented the probability error P(E)=1-P(C) for a frame to be correctly identified is 24%. When we consider a harmonic state staying stable for over a multiple frame, the probability of having an insertion within that particular frame group is exponentially increasing. This suggests that when the lenght of a harmonic state increases, we have a greater chance to incorrectly transcribe the corresponding harmonic state and this will lead us incorrectly count the length of the temporal information. Using time models, which is directly related to the counting of the stable (non-changing) states, we fail to correctly represent the time information directly from the lexical chord representation. Figure 3.16 shows the true transcription of a note harmonic state length distribution of a random song in our database and its corresponding state count model distribution. One can see that the two note distributions do not overlap and this causes the time models to fail for helping in retreival experiments. That is why we concluded at this stage of the research that, we should consider temporal information completely separately from the harmonic representation and this is set to be our main discussion for future work.

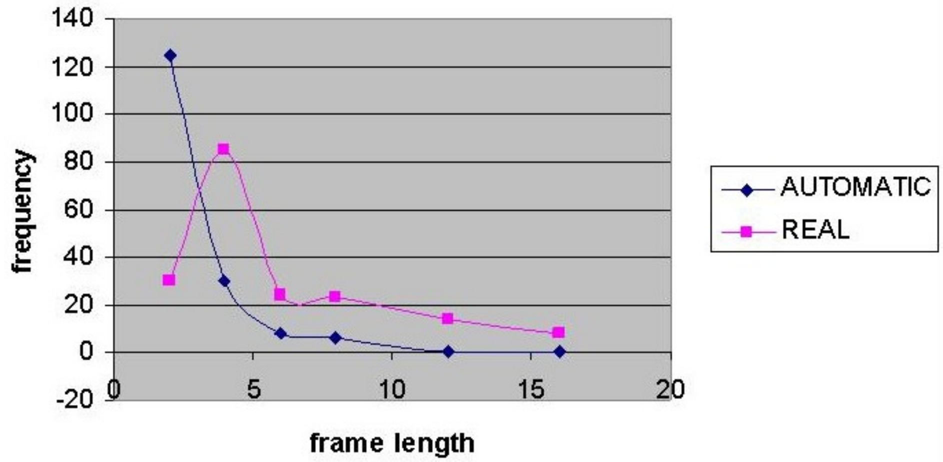


Figure 3.16: Comparison of note length distributions (automatic vs true transcriotion)

3.5 Conclusion

In this section, we have presented a mid-level representation scheme for polyphonic music audio that is independent of the note level transcription performance. Since the salient pitches form the most important features in the defining of musical chord identity, we claimed that the effect of spectral differences of different instruments can be and should be ignored. This is achieved by mapping the audio spectrum to the Spiral Array model, which accurately tracks the tonal behavior of the melodies for sequential modeling. We used perplexity analysis to understand how likely a query sequence can be generated by melodic models in a collection. Results for the retrieval tests showed that, on average, around 70.6% of the time, we can expect to retrieve a reasonable variation of the polyphonic query in the 60-best list using the mid-level lexical chord representation.

We updated the plain text representation to compansate for different tempii, by eliminating the harmonic repetitions in the representation, and used the resulting statistical models to boost the performance of the retreival system. We were able to get around 7.6% absolute retreival performance improvement in 10-best list.

We also tested the system for a more practical test case, where a short clip of a melody is used as an input to the system. In this scenerio, we also constructed generic models for each melody from its all available variations. Our retrieval experiments showed that, the system was able to retrieve the target melody with 72.1% from a dataset of 4373 music pieces (individual models for variations are concatinated into one generic model with N=45.

As future work, we plan to test our representation scheme and the retrieval system by expanding our chord vocabulary to include 7th chords. This will help the system to epresent a wider range of musical samples, and also a more detailed tracking of the harmonic movement.

A more general comparison can be achieved by testing our system with other work in the literature, and MIREX-cover song identification task competition is one of the most appropriate ways to do so, where researchers test their retrieval systems on the same dataset. Our next objective is to attend the next available competition for cover song identification task and see how well our algorithm performs against other researcher's work.

Chapter 4

Conclusion

Two main problems of Music Retrieval Systems was under discussion in this paper. We first discussed the Query by Humming Problem, and then moved to polyphonic domain and discussed our efforts for solving Query by Example problem. For both topics, we first stated the problem with tasks to be completed and their related challenges, and proposed our solution for robust performance under the effect of uncertainty.

In Chapter2 we presented a statistical approach to the problem of retrieval in the presence of uncertainty for QBH systems. Our research was motivated by the way humans perceive and reproduce music. Knowledge-based methods incorporating human understanding of music were combined with data-based models with the help of statistical signal processing techniques.

Our statistical approach required the extraction of FPs from hummed melodies which are subsequences of the input query. Our main assumption was that the subsequences, covered by the FP's was unique enough to represent the melodies that were meant to be performed. However, they were expected to carry large amounts of uncertainty.

The search for these FPs was informed by our previous findings on inter-person humming variability, calculated from a database of humming samples by both musically trained and untrained participants. Our results showed that the size of the test database and the subjects musical training are the main factors that determined the success of our approach. We also implemented an ED approach for retrieval in order to compare the performance of the proposed FP algorithm to a more conventional approach. In our comparison for different test database sizes, the FP approach consistently outperformed the ED approach, and the results showed that the FP algorithm is more robust to user-dependent uncertainty, especially for the more noisy humming queries formulated by nonmusically trained participants.

One important question that one can ask at this point is how the performance of the retrieval system is affected by the accuracy of the front-end recognizer. For instance, for queries formulated by musically trained participants, where we can expect lowlevels of user-based errors, the dominant factor that impedes system performance are likely the system-based transcription errors, such as note insertions and deletions.

Future work will include further improvements on the front-end segmentation and recognition processes in order to achieve even better retrieval results, especially addressing the issue of mixed hummed and/or sung retrieval queries. We intend to use existing databases, our own, and combinations of the two to evaluate such schemes. The development of a scalable front-end segmentation algorithm will be critical to allow for unrestricted human audio input.

Moreover, better statistical models for the retrieval side of the system can also be developed, such as by extracting repeating patterns that could represent the most likely expected patterns observed in the users humming input. This, in turn, implies higher penalties for unlikely database elements that will eventually result in improved retrieval calculations. Conversely, we could study the occurrence of rare melodic patterns as signatures to help enable fast retrieval.

In Chapter 3, we have presented a mid-level representation scheme for polyphonic music audio that is independent of the note level transcription performance. Researchers already reported achieving a high level transcription from audio to true note transcription is a hard task. For this, a mid level representation that tracks tonal behaviour rather than pitch related labels is preferred. Since the salient pitches form the most important features in the defining of musical chord identity, we claimed that the effect of spectral differences of different instruments can be ignored. This is achieved by mapping the audio spectrum to the Spiral Array model, which accurately tracks the tonal behavior of the melodies for sequential modeling. We have updated the Spiral Array with respect to its spelling related functionality, and created a 3 dimensional closed space. The terminology for creating Major and minor triads are repeated and the closed space is clustered accordingly. By calculating the tonal center of each consecutive frame in audio, and tracking it towards the clustered space, we achieved a one dimensional representation of polyphonic audio with lexical triad chords.

We used perplexity analysis to understand how likely a query sequences can be generated by melodic models in a collection. Results for the retrieval tests showed that, on average, around 70.6% of the time, we can expect to retrieve a reasonable variation of the polyphonic query in the 60-best list using the proposed mid-level lexical chord representation.

We updated the plain text representation to compansate for different tempo, by eliminating the harmonic repetitions in the representation, and used the resulting statistical models to boost the performance of the retreival system. We were able to get around 7.6% absolute retreival performance improvement in 10-best list.

We also tested the system for a more practical test case, where a short clip of a melody is used as an input to the system. In this scenerio, we also constructed generic models for each melody from its all available variations. Our retrieval experiments showed that, the system was able to retrieve the target melody with 72.1% from a dataset of 4373 music pieces (individual models for variations are concatinated into one generic model with N=45.

As future work, we plan to test our representation scheme and the retrieval system by expanding our chord vocabulary to include 7th chords. This will help the system to represent a wider range of musical samples, and also a more detailed tracking of the harmonic movement. However, the generated subsequences in the statistical sequential models will lead slower computation time for both training and testing.

One of the most important things to discuss in our future work is the tempo information. Since eliminating the tempo variation boosted our retreival performance when we use this pure harmonic information as supporting models in the perplexity calculations, the information in the time axis also should be helpful. However due to transcription error probability, this time related information can not be extracted accurately from the lexical chord representation. This is why we conclude that tempo and rhythm related information should be covered seperately from this chord representation terminology and should be calculated using more robust feature sets such as frame level energy. The feature set that we used in building harmonic models was dependent on fft information and this information might not necessarily be usefull in finding meaninfull tempo marks thus the rhythm. We strongly believe that rhythmic characteristic should be incorparated in retrieval since the characteristic information that is carried by temporal information is vital for identifying similiar melodies from a dataset. This information can be used directly in retrieval by using either emplying string matching type of distance measurement or indirectly by using it in clustering the database for grouping similiar songs that has common rhythmic characteristics. Clustering approach may decrease the search time and computation, in the expense of extracting accurate rhythmic information, which is a very complex research area.

References

[BR98] S. Blackburn and D. De Roure. A tool for content based navigation of music. In *Proc. ACM International Conference on Multimedia*, pages 361–368, Bristol, England, September 1998.

[Cap] B. Capleton. Perfect pitch.

[CC05] C.-H. Chuan and E. Chew. Polyphonic Audio Key-Finding Using the Spiral Array CEG Algorithm. In *Proc. IEEE-ICME International Conference on Multimedia and Expo*, Amsterdam, Netherlands, 2005.

[Che00] E. Chew. *Towards a Mathematical Model of Tonality*. PhD thesis, Massachusetts Insitute of Technology, Cambridge, MA, 2000.

[Che01] E. Chew. Modeling Tonality: Applications to Music Cognition. In *Proc. CogSci2001 23rd Annual Meeting of the Cognitive Science Society*, Edinburg, Scotland, 2001.

[CJ98] B. Chen and J.-S Roger Jang. Query by singing. In *Proc. IPPR Conference on Computer Vision, Graphics and Image Processing*, pages 529–536, Taiwan, August 1998.

[CML+02] L. P. Clarisse, J. P. Martens, M. Lesaffre, B. De Baets amd H. De Meyer, and M. Leman. An auditory model based transcriber of singing sequencecs. In *Proc. ISMIR International Conference on Music Information Retrieval*, Paris, France, October 2002.

[DBT+03] R. B. Dannenberg, W. P. Birmingham, G. Tzanetakis, C. Meek, N. Hu, and B. Pardo. The musart testbed for query-by-humming evaluation. In *Proc. ISMIR International Conference on Music Information Retrieval*, Baltimore, MD, October 2003.

[DEKM98] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Allignmnet*. Cambridge University Press, Cambridge, UK, 1998.

[DR02] S. Doraisamy and S. Ruger. A Comparative and Fault-tolerance Study of the Use of n-grams with Polyphonic Music. In *Proc. International Conference on Music Information Retrieval*, Paris, France, 2002.

[ECM08] D. P. W. Ellis, C. V. Cotton, and M. I. Mandel. Cross-correlation of beat-synchronous representations for music similarity. In *Proc. ICASSP International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, NV, 2008.

[EU08] Elaine Chew Shrikanth Narayanan Erdem Unal, Panayiotis Georgiou. Statistical modeling and retrieval of polyphonic audio. In *Proc. IEEE Multimedia Signal Processing*, Crete, Greece, 2008.

[Fit54] P.M. Fitts. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47:381–391, 1954.

[GLCS95] A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith. Query by humming: Musical information retrieval in an audio database. In *Proc. ACM International Conference on Multimedia*, pages 231–236, San Fransisco, California, November 1995.

[HD02] Ning Hu and Roger B. Dannenberg. A comparison of melodic database retrieval techniques. In *Proc. ACM International Conference on Digital Libraries*, Portland, Oregon, July 2002.

[HHMS99] K. Honda, H. Hirai, S. Masaki, and Y. Shimada. Role of vertical larynx movement and cervical lordosis in f0 control. *Language and Speech*, 42:401–411, 1999.

[IHSM04] A. Ito, S.-P. Heo, M. Suzuki, and S. Makino. Comparison of features for dp-matching based query-by-humming systems. In *Proc. ISMIR International Conference on Music Information Retrieval*, Barcelona, Spain, October 2004.

[KNS+99] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamuro, and K. Kushima. Music retrieval by humming - using similarity retrieval over high dimensional feature vector space -. In *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, pages 404–407, Victoria, Canada, August 1999.

[LP03] V. Lavrenko and J. Pickens. Polyphonic Music Modeling with Random Fields. In *Proc. ACM Multimedia 2003*, Berkeley, CA, 2003.

[Mar] M. Marolt. A mid level melody based representation for calculating audio similarity.

[MC05] A. Mardirossian and E. Chew. Key Distributions as Musical Fingerprints for Similarity Assessment. In *Proc. IEEE-MIPR International Workshop on Multimedia Information Processing and Retrieval*, Irvine,CA, 2005.

[MSW+96] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham. Towards the digital music library: Tune retrieval from acoustic input. In *Proc. ACM International Conference on Digital Libraries*, pages 11–18, Bethesda, Maryland, March 1996.

[MSWH00] R. J. McNab, L. A. Smith, I. H. Witten, and C. L. Henderson. Tune retrieval in multimedia library. *Multimedia Tools And Applications*, 10:113–132, April 2000.

[Par05] Charles Parker. Applications of binary classification and adaptive boosting to the query-by-humming problem. In *Proc. ISMIR International Conference on Music Information Retrieval*, London, UK, September 2005.

[Pau02] Steffen Pauws. Cubyhum: A fully operational query by humming system. In *Proc. ISMIR International Conference on Music Information Retrieval*, Paris, France, October 2002.

[PB92] D.-B. Paul and J.-M. Baker. The design for the wall street journal-based csr corpus. In *Proc. HLT of the workshop on Speech and Natural Language*, Harriman, New York, 1992.

[PBC+02] J. Pickens, J. Bello, T. Crawford, M. Dovey, G. Monti, and M. Sandler. Polyphonic score retrieval using polyphonic audio queries: A harmonic modeling appriach. In *Proc. ISMIR International Conference on Music Uinformation Retrieval*, 2002.

[PE07] G. E. Poliner and D. P. W. Ellis. A Discriminative Model for Polyphonic Piano Transcription. *EURASIP Journal on Advances in Signal Processing*, 2007, 2007.

[PI03] A. Pertusa and J. M. Inesta. Polyphonic Music Transcription Through Dynamic Networks and Spectral Pattern Identification. In *Proc. IAPR International Workshop on Artificial Neural Networks in Pattern Recognition*, Florence, Italy, 2003.

[PMS+03] J. Pickens, J.P. Belloand G. Monti, M. Sandler, T. Crawford, M. Dovey, and D. Byrd. Polyphonic Score Retrieval Using Polyphonic Audio Queries: A Harmonic Modeling Approach. *Journal of New Music Research*, 32, 2003.

[PSB04] B. Pardo, J. Shifrin, and W. Birmingham. Name that tune: A pilot study in finding a melody from a sung query. *Journal of the American Society for Information Science and Technology*, 55:283–300, February 2004.

[Rap02] C. Raphael. Automatic Transcription of Piano Music. In *Proc. ISMIR International Conference on Music Information Retrieval*, Paris, France, 2002.

[RK05] M. Ryyananen and A. Klapuri. Polyphonic Music Transcription Using Note Event Modeling. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2005.

[Ros02] T. D. Rossing. *Science of Sound*. Addison-Wesley, San Francisco, CA, 2002.

[RRG99] P. Y. Rolland, G. Raskinis, and J. G. Ganascia. Music content-based retrieval: An overview of melodiscov approach and systems. In *Proc. ACM International Conference on Multimedia*, pages 81–84, Orlando, Florida, November 1999.

[SB03] P. Smargadis and J. C. Brown. Non-Negative Matrix Factorization for Polyphonic Music Transcription. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, NY, 2003.

[SG08] Joan Serra and Emilia Gomez. Audio cover song identification based on tonal sequence alignment. In *Proc. ICASSP International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, NV, 2008.

[SNK02] H.-H. Shih, S. Narayanan, and C.-C. J. Kuo. An hmm-based approach to humming transcription. In *Proc. IEEE International Conference on Multimedia and Expo*, pages 337–340, Laussanne, Switzerland, August 2002.

[SNK03] H.-H. Shih, S. Narayanan, and C.-C. J. Kuo. Multidimensional humming transcription using hidden markov models for query by humming systems. In *Proc. IEEE International Conference on Acoustics Speech and Signal Processing*, Hong Kong, April 2003.

[Sto02] A. Stolcke. Srilm – an Extensible Language Modeling Toolkit. In *Proc. International Conference on Spoken Language Processing*, Denver, CO, 2002.

[SUN08] K. Samuel, Erdem Unal, and Shrikanth Narayanan. Fingerprint extraction for classical music cover song identification. In *Proc. IEEE-ICME International Conference on Multimedia Expo*, Hannover, Germany, 2008.

[TYW02] W.-H. Tsai, H.-M. Yu, and H.-M. Wang. Query by example technique for retrieving cover versions of popular songs with similiar melodies. In *Proc. ISMIR International Conference on Music Information Retrieval*, 2002.

[UCGN08] E. Unal, Elaine Chew, Panayiotis Georgiou, and Shrikanth Narayanan. Challenging uncertainty in query-by-humming systems: A fingerprinting approach. *IEEE Transactions on Audio, Speech and Language Processing, Special Issue in Music Information Retrieval*, 16:359–371, February 2008.

[UGCN07] E. Unal, Panayiotis Georgiou, Elaine Chew, and Shrikanth Narayanan. Statistical modeling and retrieval of polyphonic music. In *Proc. IEEE Multimedia Signal Processing Workshop*, Chaine, Greece, October 2007.

[Uit03] A. L. Uitdenbogerd. Was parsons right? an experiment in usability of music representations for melody-based music retrieval. In *Proc. ISMIR International Conference on Music Information Retrieval*, Baltimore, MD, October 2003.

[UNC04] E. Unal, S. Narayanan, and Elaine Chew. A statistical approach to retrieval under user-dependent uncertainty in query by humming systems. In *Proc. ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 113–118, New York, New York, October 2004.

[UNC+05] E. Unal, S. Narayanan, Elaine Chew, H.-H. Shih, and C.-C. J. Kuo. Creating data resources for designing user-centric front-ends for query by humming systems. *Multimedia Systems Journal*, 10:475–483, May 2005.

[UNC+06] E. Unal, S. Narayanan, Elaine Chew, P. Georgiou, and N. Dahlin. A dictionary based approach for robust and syllable-independent audio input transcription for query by humming systems. In *Proc. 1st ACM workshop on Audio and Music Computing for Multimedia*, pages 37–44, Santa Barbara, California, October 2006.

[vR79] C. J. van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.

[ZS03] Y. Zhu and D. Shasha. Warping indexes with envelope transforms for query by humming. In *Proc. ACM SIGMOD International Conference on Management of Data*, San Diego, California, June 2003.