International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 9, September 2012)

QBH System Using Melody Matching Model based on a String Pattern Matching mechanism of Frequency Contours

[1]K.P.P.S. Warnaweera,[2]D.K. Withanage
[1]*Undergraduate, Faculty of Information Technology, University of Moratuwa, Sri Lanka.*
Email:-p.waranaweera@gmail.com
[2]*Head, Department of Information Technology, Faculty of Information Technology, University of Moratuwa, Sri Lanka.*
Email:-withanage@itfac.mrt.ac.lk

***Abstract*— Query by humming (QBH) refers to music information retrieval systems where short audio clips of singing or humming act as queries. Melody is considered as the most important feature in the queries and the songs. This system proposes a QBH system Using Melody Matching Model Based on a string pattern matching mechanism of frequency contours, which is used to align the input pitch frequency contour with the original music pitch frequency contour. The proposing comparison algorithm provides about of 60% of accuracy which is comparatively good considering the accuracy level of existing QBH systems.**

***Index Terms*— Voice Filtering, Feature Extraction, Quantization, Similarity Measuring, Shift Edit Distance.**

I. INTRODUCTION

Music is a universal language. Music originated and evolved with developing demands of the human and the rapid discoveries in science and technology. Sharing music over World Wide Web is an upcoming trend among the computer users. Simply, music database owners use the web to fulfill the needs of the people who are frustrated with searching for their favorite music over different methods. Occasionally, fewer music search engines provide enough facilities to find the piece of music that the user requires. It is obvious that one cannot design a search engine with all the user requirements. However, it makes inconvenient for music lovers to search music by reminding metadata of a song such as song title, author name, lyrics, etc. Melody can be considered as the most quickly capturing and reminding feature of a song because human brain can easily capture the melody rather than reminding metadata of a song. Therefore, Query By Humming (QBH) refers to a robust music information retrieval system in which short audio clips of singing or humming act as queries.

This paper focuses on proposing an approach for developing a robust QBH system using a melody matching model based on a string pattern matching mechanism of frequency contours. Section 2 provides an overview of the QBH systems including its general design and existing applications. Section 3 is outlines the overall design and implementation of the proposing system including precise descriptions of the major components of the proposing system. Section 4 focuses on the future directions of QBH systems including the author's idea regarding the ways new research can take place for enhancing the accuracy of QBH systems. Section 5 is the Acknowledgement.

II. OVERVIEW OF QBH SYSTEMS

In the World Wide Web (WWW), different music from different cultures is stored and shared. The challenge for music lovers is searching for the required song among the stored music available in the WWW. As a robust solution for that challenge, QBH systems are introduced which can take audio clips as searching queries.

A. Why Search by Humming?

When finding different cultural songs in different languages it is inconvenient for music lovers to remember the Meta data regarding the song that they want to search. Moreover, normally the human can remember the melodies and rhythms of songs rather than their Meta data. Also some music listeners might have remembered only a portion of a song that they desired. To fulfill such requirements, developers are still doing research to build a proper music information retrieval system with audio input. To achieve this goal designers have to face many challengers because the accents of the different people are different from each other. Also the users may hum a song in different pitches rather than the original pitch of the song or music. The optimizing process of the searching algorithm is main challenge for developer of QBH system.

B. Existing Systems

Types of music searching techniques available today can be categorized as follows [1].

- Search by Metadata for Music Related Information.
- Search for Lyrics
- Search for Recommend Similar Music
- Search by Singing or Humming: Example for some of this type of music search engines are "Midomi" [2] which produces fairly good results compared with other QBH systems, "Tunebot", "Sound Hound" [3] and "Musipedia".

C. General Design of QBH Systems

Generally, a QBH system consists of main three modules namely

- Data Pre-Processing
- Symbolic Representation
- Comparison

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 9, September 2012)

In the Pre-Processing module, audio signal is preprocessed to implement the Feature Extraction by using errorless input wave. In Symbolic Representation module, the preprocessed features are prepared in a suitable way to be used within the Comparison component. In Comparison module, a proper comparison algorithm should be implemented to compare the symbolic representations of both the input hummed query and the original. Implementing this component is more critical as comparison should be done in a way that gives a high accurate output results.

III. The Proposed QBH System

A. Major Modules

The proposed QBH System Using Melody Matching Model Based on a String Pattern Matching Mechanism of Frequency Contours consists of three major Modules:

- Pre-Processing: Use filtering mechanism to filter instrumental parts of the original song as well as other noise to some extent.
- Indexing: Mainly consists of two sub modules namely Feature Extraction and Quantization which will be described comprehensively in the 3.3 section.
- Comparison: Mainly consists of two activities namely Similarity Measuring and Ranking which will be described comprehensively in the 3.3 section.

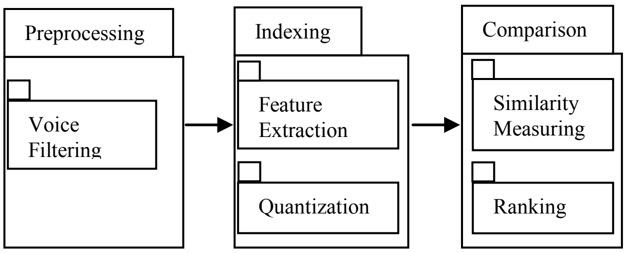

Fig. 1. Component Diagram of the proposing QBH System.

B. The High-level Design

Initially, the voice filtering and Feature extraction processes are performed for the original songs in the musical database (DB) and the extracted frequencies of each song are stored in the musical DB accordingly for each and every song. When a user inputs the humming or singing audio query it will be recorded and then it will be processed with the indexing component where, feature extraction and quantization processes are performed.

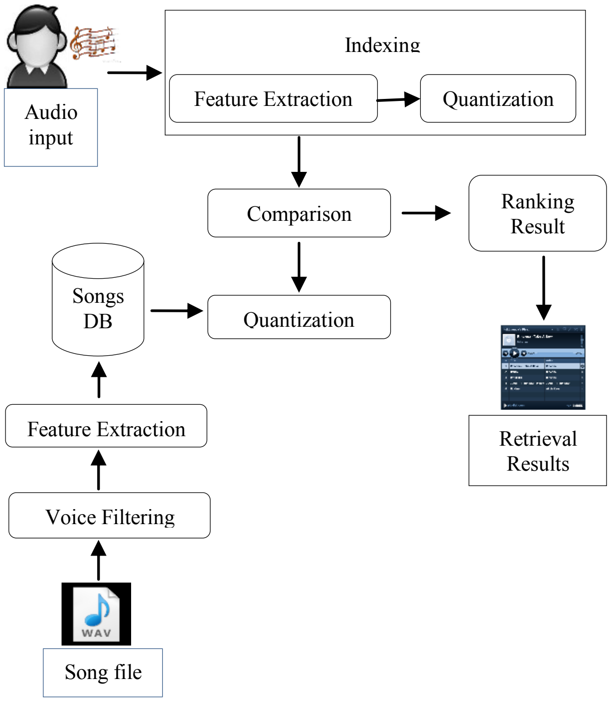

Fig. 2. High Level Design of the proposing QBH System

In feature extraction, frequencies are extracted from the audio singing/humming query and quantization process generates a symbolic representation from the frequencies for implementing the melody matching in an accurate manner. Once input query is processed, quantization takes place for stored frequency sets of the original songs in the DB. Then by the comparison process, the comparison will take place between symbolic representations of both the input query and the target song in the DB. Then it ranks each song according to the comparison results and display the ranking results as a list of songs.

C. Implementation of the Proposed System

The implementation of major three modules of the proposed system namely Preprocessing, Indexing and Comparison are described below.

1.) Pre-Processing:-

The process of searching is performed initially by a humming input generated vocally by a potential user. Although an original song consist both voice and the instrumental part, the system uses only its pitch of the vocal part. This system uses band filtering and channel filtering methods to remove instrumental parts of the song. One advantage of doing so is to remove the empty gap between recoding at the start and end as well. Thus the system performs filtering at the initial, and then we can remove the beginning and the end empty gap of recordings. Most of currently available QBH systems follow a similar method

IJETAE
EXPLORING RESEARCH AND INNOVATIONS

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 9, September 2012)

2.) Indexing:-

This module mainly consists of two sub modules namely Feature Extraction and Quantization.

Feature Extraction: -

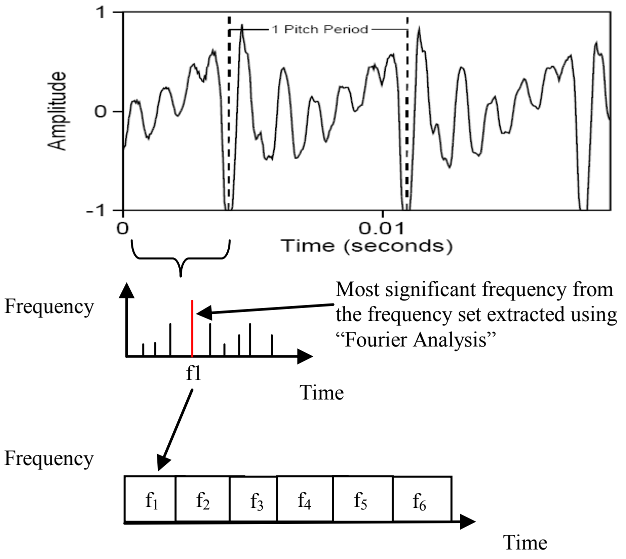

Fig. 3. High Level Design of the proposing QBH System

Both original (stored in the database) and humming signals of the song in the frequency domain are sampled (framed) at regular time intervals and the most significant frequency is extracted after “Fourier Analysis” of the sample. For each sample, this is repeated. Most of currently available QBH systems follow a similar methodology. The main factor to consider in framing is the frame size (The time period of each frame). We have to consider about the human pitch resolution [4] before deciding the frame size. Most of the research papers have stated that the time period which is suitable for framing while keeping the human pitch resolution factor is 20- 23 ms [6] [7]. Using this set of frequencies, melody contour is generated for each audio. These frequency sets for each and every song are stored in the musical database accordingly.

Quantization: -

In this module, the extracted frequencies are quantized and represented by sequence of symbols of the English Alphabet. Here, the extracted frequencies are quantized in to intervals and each interval is represented by a specific symbol. Non linier quantization is used to obtain a high accuracy. That symbol is considered as a bin to store frequencies within a specific range if frequencies. This process is represented by generating a look up table to represent assigned symbols for each frequency range.

Here, the important parameters are,

- Size of the alphabet
- Corresponding interval width (frequency range)

To determine these parameters, we consider the lowest and highest frequencies of the frequency set. To implement quantization the frequency set a song or an audio query should be divided in to equally sized fragments. The size is determined by the size of the audio. The quantization takes place for fragments individually along with the input query. Here, even though, works with fragments is slows the comparison algorithm, it is more precise. By this process, each frequency is represented by a string symbol which can be considered as a bin.

0 - 10	A
11 - 20	B
21 - 30	C
31 - 40	D

Quantization

{11, 24, 6, 15, 9, 35} ⟹ {B, C, A, B, A, D}

Fig. 4. Look-up table

This symbolic representation makes the similarity matching algorithm easier as here the series make more legible and this normalize the target song and the query in to an absolute value. These symbolic representations of both the input query and target song are the output for the Comparison component of the proposing system.

3.) Comparison:-

Comparison module carries over two activities namely, Similarity measuring and Ranking. Similarity Measuring is implemented by using the Shift Edit Distance measuring. [8] The Shift Edit Distance is based on “Levenshtein distance” which is used to measure the difference between two strings. This process is repeated for each and every fragment of a query. The ranking of the songs is processed according to a specific value calculated by string distance measuring.

Similarity Measuring:-

This process is responsible for determining the target fragment of the target potential song with the closest shape of the contour of the audio query. In this case, the frequency contour of the input query is slid through the frequency contour of the target song to find the best match. Each candidate song is given a value to determine the rank of that song within the list of songs which will be given as the output.

To implement this, use “Shift Edit Distance” which is based on “Levenshtein Distance”. Here, the QBH system takes each fragment of the target song and measures the amount of difference between the string sequences of the considered fragment of the target song and the input query. To determine this, a cost function uses to determine the cost to replace two strings characters. Here, the cost is calculated by, calculating the distance between X and Y symbols and divided by the size of the alphabet being used. Here, the distance(X,Y) is the difference between the two symbols.

$$\text{cost}(X,Y) = \text{dist}(X,Y) / \text{Size of the alphabet}$$
$$\text{dist}(X,Y) = |\text{index}(X)-\text{index}(Y)|$$

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, Volume 2, Issue 9, September 2012)

The "Shift Edit Distance" takes place at the beginning of the fragment by shifting one position to the right in a way all possible parts of the target fragment are compared with the query. Here, the distance between the input query and the target fragment is calculated by getting the sum of the letter by letter cost. After calculation the total cost/distance for each comparison or position, the minimal cost of each comparison/position takes places is selected for that particular fragment and it is considered as the final cost for that fragment. By following this process, costs for each and every fragment are determined. The minimal fragment cost (distance) is considered as the overall cost of that target song.

Ranking:-

According to the value of overall distance of each and every song calculated within the Similarity Measuring process, ranking is determined.

IV. Conclusion

The evaluation of the results of the proposed system was done using music database of three different number of song sets. First song set consisted with twenty different songs and in the result list of songs; the expected song was ranked as the seventh. Second set of songs consisted with forty songs and in the result song list; the expected song was ranked as the fifteenth. The third set of result consisted with sixty songs and in the result song list; the expected song was ranked as the twenty first. Therefore, by evaluating the results, the overall system accuracy (searching accuracy of a song) can be roughly calculated as 60%.

In future, a more robust system which enables to filter back ground noise effectively can be developed. In addition, finding a more efficient algorithmic process to implement melody matching seems a very promising future development.

Acknowledgment

The authors would like to thank for the support provided by the staff of the Faculty of Information Technology, University of Moratuwa, Sri Lanka.

References

[1] Alexandros Nanopoulos, Dimitrios Rafailidis, Maria M. Ruxanda and Yannis Manolopoulos. (2009, March. 19). Music Search Engines: Specifications and challenges [Online].Available: http://www.sciencedirect.com/.

[2] midomi. midomi. URL: http://http://www.midomi.com/.

[3] soundhound. soundhound. URL:http://www.soundhound.com/.

[4] Psychoacoustics - Wikipedia, the free encyclopedia. Psychoacoustics [Online]. Available: http://en.wikipedia.org/wiki/Psychoacoustics

[5] Russ Rowlett. (2001, July. 13). A Directory of Unit of Measurement [Online]. Available: http://www.unc.edu/~rowlett/units/dictC.html.

[6] Lloyd A. Smith, Rodger J. McNab, and Ian H. Witten. Music Information Retrieval Using Audio Input [Online]. Available: http://eprints.kfupm.edu.sa/52629/1/52629.pdf.

[7] Arunan Ramalingam and Sridhar (Sri) Krishnan. (2006, December). Gaussian Mixture Modeling of Short-Time Fourier Transform Features for Audio Fingerprinting [online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&isnumber=&arnumber=4014108

[8] Alexander Duda: Query by singing/humming with low-level feature extraction.