Presented at National Conference on Communications, NCC 2003, Jan 31 –Feb 2, IIT Madras 2003

TANSEN: A QUERY-BY-HUMMING BASED MUSIC RETRIEVAL SYSTEM

M. Anand Raju, Bharat Sundaram and Preeti Rao*

Department of Electrical Engineering,
Indian Institute of Technology, Bombay
Powai, Mumbai 400076
{maji,prao}@ee.iitb.ac.in *Dept. of EE, I.I.T. Kanpur

ABSTRACT

Music information retrieval is a field of rapidly growing commercial interest. This paper describes TANSEN, a query-by-humming based music retrieval system under development at IIT, Bombay. Named after the legendary musician (and a tenuous acronym for "TA-Note Song Extractor-Navigator), the system is designed to accept acoustic queries in the form of sung fragments, to search a database of Indian film songs. Algorithms for the extraction of melody from the query signal, and pattern matching for search and retrieval from the database are presented. The user interface is described, and experimental results obtained on a prototype version are reported.

1 INTRODUCTION

Digital representations of music are becoming common for the storage and transfer of music over Internet. Many digital music archives are now available, making the content based retrieval of music a potentially powerful technology. The recent MPEG-7 audio standardization activity [1] seeks to develop tools for the description and intelligent searching of audio content. Searching for music based on tune or melody is an important component of any content retrieval system that targets music databases. While melody is only one of many aspects of a piece of music, it is certainly among its most salient features. This is especially true of songs (vocal music). For example, the most natural way of querying a database of songs would be by humming a fragment of the desired song. Query-by-humming (QBH) is therefore an important application within the scope of MPEG-7. A melody retrieval system based on acoustic querying would allow a user to hum or sing a short fragment of a song into a microphone and then search and retrieve the "best matched" song from the database.

This paper presents TANSEN, a query-by-humming music indexing and retrieval system based on melody, or the "tune", of the music. An earlier paper [2], written during the starting phase of this project, introduced the basic functional blocks and outlined the challenging problems posed by this application.

Figure 1 shows the functional blocks of a basic melody based retrieval system. The melody database is essentially an indexed set of soundtracks. The acoustic query, which is typically a few notes whistled, hummed or sung by the user (presently restricted to the syllable "ta" for reasons explained later), is processed to detect its melody line. The database is searched to find those songs that best match the query. The system returns a ranked set of matching melodies, which can be used to retrieve the desired original soundtrack. The major algorithmic modules therefore are the extraction of a melody representation from the query (and also the database songs at the time of creating the database), and the melodic similarity distance computation.

While the overall task is one that is easily performed by humans, many challenging problems arise in the implementation of an automatic system. These include the signal processing needed for extracting the melody from the stored audio and from the acoustic query, and the pattern matching algorithms to achieve proper ranked retrieval. Further, a robust system must be able to account for inaccuracies in the user's singing. The system will typically operate on a substantial database and must respond within seconds.

The recent growth of interest in melody retrieval research is evident by the efforts of major audio research groups including MIT Media Labs [3],

Cornell University [4] and Waikato Univ. in New Zealand [5]. The New Zealand group has developed a prototype system (known as MELDEX) with a folk song database of 10000 public domain songs. MELDEX uses a 3-level pitch contour and rhythm information to represent melody. In this system, the first 20 notes of the query are considered. Dynamic programming is used for searching. 'Tuneserver', developed [6] at the University of Karlsruhe in Germany, has a database of 10000 classical, 100 popular, 15000 folk songs and 100 national anthems. Here 3-level pitch contour is used to represent melody. Whistling is the only form of querying supported. The University of Bonn audio group is also working on a QBH system [7] known as 'MiDiLib'.It has database of 2000 MIDI files. This group uses a greater than 3 level pitch contour representation along with rhythm. The LCS (longest common subsequence) algorithm is used for matching. Whistling is a query input.

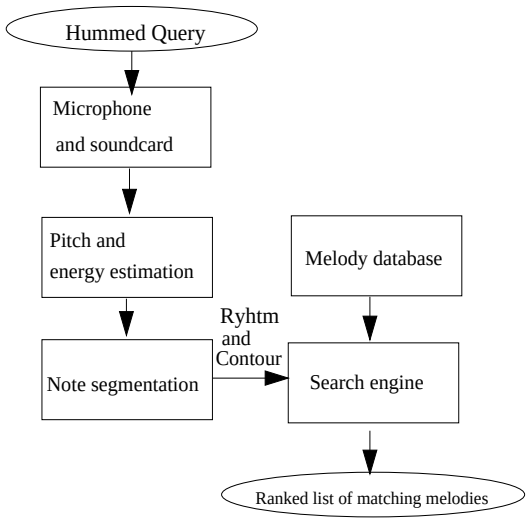

Figure 1. A melody based music retrieval system.

Building an effective music retrieval system, we believe, requires an appreciation of the characteristics of the music database that is targeted. The melody representation scheme and the string matching algorithm that are chosen must capture the distinctiveness of the member items and also reflect accepted notions of melodic similarity. Further it is important to account for the typical inaccuracies in user queries as obtained from realistic field studies.

Our system is intended for a database of Indian music, in particular, Hindi and regional film songs. This musical 'genre' (if it may be called so inspite of its mix of Indian classical, traditional and, more recently, Western influences) enjoys tremendous popularity with a wide appeal that transcends nearly all geographical, language and social barriers in India. That Indian film music has a strong Internet presence is borne out by the number of websites that offer film song sound tracks for downloading, often searchable by composer, singer or lyrics.

2. MELODY REPRESENTATION

The fundamental attributes of music are the pitch sequence of notes, rhythm, tempo (slow/fast), dynamics (loud/soft), texture (timbre or voices) and lyrics (if any). It is in these dimensions that we typically distinguish one piece of music from another. Of these descriptors, melody and rhythm are the most distinctive. The melody of a piece of music is a sequence of notes with varying pitch and duration. The pitch is associated with the periodicity of the sound, and allows the arranging of sounds ranked low to high on a musical scale. What we perceive in music is not only the pitch of individual notes but also how they correspond to particular moments in time, which is described by the rhythm attribute.

Although the melody is described by the time sequence of pitches, it is evident that people are able to recognize melodies even after pitch transposition (as the same tune played in a different key). For this reason, more characteristic than the absolute pitches of the successive notes are the relative frequency intervals between the notes. This relative variation of pitch in time is known as the "pitch contour", and it provides a dimension which is invariant to key transposition. Apart from pitch contour, the only other dimension in which melodies in general cannot be transformed is the rhythm [3]. There has been research on how music is remembered. Dowling [8] discovered that the melody contour is easier to remember than exact melodies. Contour refers to the shape of the melody, indicating whether the next note goes up, down, or remains at the same pitch

Various representations for melody have been proposed: (i) Pitch contour representation: 3-level (U/D/S indicating that the pitch goes up, down, or

remains the same) [8], or 5-level (++/+/0/-/--) (ii) Pitch contour with duration representation: along with 3-level pitch contour, each note duration is also specified. (iii) Absolute pitch representation: a melody is converted into a normalized pitch sequence by mapping the pitches into one octave from C4 to B4, i.e. there is a total of 12 symbols. Currently, for simplicity and robustness to query inaccuracies, we adopt the 3-level pitch contour without rhythm information. That is, the query signal is segmented into distinct notes, each of which is assigned a pitch value in Hz. Next the U/D/S string is obtained from comparing the pitch values of every two successive notes.

3. PROCESSING THE QUERY

From the previous section we see that reliable note segmentation is a critical aspect of query processing. In order to simplify note segmentation, we currently require that the query be sung using a syllable such as "ta". The stop consonant "t" causes the local energy of the waveform to dip thus making for relatively easy identification of note boundaries. We compute the instantaneous energy of query waveform averaged over 25 ms frames. This energy contour requires smoothing because energy spikes are created due to improper recording, stray mic clicks etc. It is done using simple median filtering. The note on/off threshold is set adaptively to adjust for any ambient noise while recording.

There exist several algorithms for detecting the pitch of an acoustic signal [9]. We have used time domain autocorrelation function for pitch extraction since it computationally simple and fast. It is computed on non-overlapping frames of fixed duration (equal to 3 times the lowest expected pitch period). Fig. 3 shows an example waveform with the energy and pitch contours. Labeling the pitch with a musical note name may seem a simple operation, but mapping frequency (which is continuous) onto the musical scale (which is discrete) causes problems because the pitch within a given note may vary over its duration. It has been observed from experiments that people who are not trained in music tend vary their pitch during a note to a large extent unknowingly. Therefore a pitch smoothing operation is necessary to assign a single pitch value to each note. This is achieved by an (empirically derived) algorithm that averages pitch values within the 50% to 80% duration range of the note.

4. STRING MATCHING FOR MELODY RETRIEVAL

The database is a set of songs indexed by the melody string of the signature phrase (or the most easily recalled phrase) of the song. Extracting the melody representation from the original soundtrack is a difficult problem that is addressed separately in an accompanying paper [10]. Currently, we obtain "model" queries from a trained singer and use these to obtain the melody representation for the database songs.

User queries cannot be expected to be completely accurate with respect to the actual pitch contour of the desired music. Typical inaccuracies are [11]: (i) insertion of new notes (ii) replacement by different note (iii) deletion of notes. These inaccuracies can be taken care of by a dynamic programming (DP) based "edit distance" algorithm [11]. DP is used to obtain minimum edit distance between two sequences. If minimum edit distance between two sequences is 0, then it is an exact match. If the minimum distance is high, then the sequences are considered to be very dissimilar. DP algorithm is given as: Let a $= (a_1, a_2, \ldots\ldots a_m)$ be a sequence of notes of a string A, each of which is encoded as a pitch change direction and b = $(b_1, b_2, \ldots\ldots b_n)$ be another sequence of notes of string B. We compute the edit distance $d_{A,B}$ of the two sequences a and b recursively as follows:

$$d_{ij} = \min \begin{cases} d_{i-1,j} + w(a_i, 0) \text{ (deletion)} \\ d_{i-1,j-1} + w(a_i, b_j) \text{ (match/change)} \\ d_{i,j-1} + w(0, b_j) \text{ (insertion)} \end{cases}$$

The initial conditions are:

$$d_{0,0} = 0$$
$$d_{i,0} = d_{i-1,0} + w(a_i, 0), i \geq 1$$
$$d_{0,j} = d_{0,j-1} + w(0, b_j), j \geq 1$$

where $w(a_i, 0)$ is the weight associated with the deletion of a_i, $w(0, b_j)$ is the weight for insertion of b_j, and $w(a_i, b_j)$ is the weight for replacement of element i of sequence A by element j of sequence

B. The operation titled *"match/change"* sets *w (ai, bj)* = 0 if *ai* = *bj* and a value greater than 0 if $ai \neq b_j$. The weights used here are 1 for insertion, deletion and substitution(change) and 0 for match. As an example, if two pitch contour strings *UDDSSUD and *UDDSUD are compared , the edit distance is 1. It is evident from the optimal alignment shown in Figure 2.

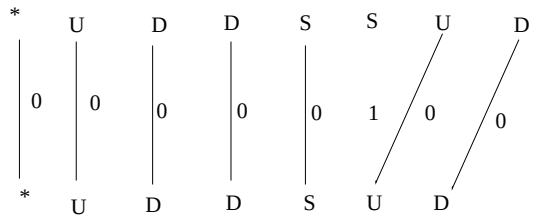

Figure 2. Optimal alignment of two strings with an edit distance = 1

5. THE USER INTERFACE

It is intended to have a web-enabled user interface to TANSEN. Based on currently available technology, it is possible to upload a previously recorded audio input file, do the required query signal processing (either on the client side or server side), and use the generated text string to search an indexed database of songs on the server. Finally, the first three best matched songs are returned by means of links to the corresponding audio soundtracks as shown in the sample output page of Fig. 4. Also displayed is the pitch contour obtained from the user query. (We plan to enhance this with a plot of the actual pitch contour of the best matched song from the database. This has the interesting potential to serve as a valuable instructional tool.)

To implement the desired user interface, file upload, http response writing, we could have used either CGI or Java Servlets running on an http server. Servlets were chosen because of their superior performance, ability to effectively handle multiple requests, portability of the code and better security. Java Servlets can be run on any Java enabled server supporting servlets. We have implemented the TANSEN user interface on the server included with JSDK2.1 which is a simple multithreaded server.

The server was installed and run from a Windows 2000 platform. The client-side operations are: recording of the query to a standard audio format; and uploading this query file. The server-side operations are: reading the uploaded file at the server; query signal processing of the uploaded file; displaying the pitch contour; searching the indexed database; printing the ranked matches on the client's page.

6. EXPERIMENTAL RESULTS AND FUTURE WORK

A small prototype system has been implemented with a database of 20 well-known Hindi film songs. The songs are indexed by the U/D/S pitch contour of the signature phrase of the song. The user is expected to sing (with syllable "ta") the signature phrase of the desired song. The acoustic query signal is recorded in mono through a microphone and PC sound card with sampling rate 22.05 kHz and 16-bit resolution.

Five users (none of whom were trained singers) were asked to provide a query for each of the 20 songs thus generating an experimental data set of 100 queries. Table 1 summarises the results of this experiment which showed a 95% success rate. "Mismatch" indicates a wrong best match. "Conflict" indicates that along with the correct match, one or more additional songs qualified with the identical similarity distance. A close analysis revealed that most cases of mismatch and conflict were due to large (and obvious) inaccuracies in the user query. Apart from this formal experiment, the system has been tested informally by a large number of people and has shown a high degree of robustness.

Of immediate importance is increasing the number of songs in the database. This work is underway, and it is expected that a convincing demo on a realistic database will be presented at the Conference. Only with a database of at least a few hundred songs can issues of what is the best melody representation and similarity distance method be addressed satisfactorily. The complexity of searching a large database must also be considered. It is expected that including rhythm in the melody representation will improve performance in terms of reducing conflicts and mismatches. This will require research on a rhythm detection algorithm.

Database Songs	**20**
Queries	**100**
Mismatch	**5**
Conflicts	**22**
Success rate	**95%**

Table 1. Summary of experimental results

5. REFERENCES

[1] MPEG-7, http://mpeg.telecomitalialab.com/standards/mpeg-7/mpeg-7.htm

[2] M.Anand Raju, Preeti Rao, "Building a melody retrieval system", *Proc.NCC*, Mumbai, Jan 2002

[3] Kim.Y.E, Chai.W, Garcia.R, Vercoe.B, " Analysis of a contour-based representation for melody ", *Proc. International Symposium on Music Information Retrieval*, Oct 2000.

[4] Ghias A, Logan J, Chamberlin D, Smith B.C, " Query By Humming ", *Proc. ACM Multimedia*, San Francisco, 1995

[5] McNab.R.J, Smith.L.A, Witten.I.H, Henderson.C.L, Cunningham.S.J, " Towards the Digital Music Library: Tune retrieval from acoustic input", *Proc. ACM Digital Librarie*s, 1996.

[6] Tuneserver, http://tuneserver.de

[7] MiDiLib, http://www-mmdb.iai.uni-bonn.de/forschungprojekte/midilib/english/

[8] Dowling.W.J, "Scaling and contour:Two components of a theory of memory for melodies", Pshychological Review, vol.85,no.4, pp.341-354, 1978.

[9] Rabiner.L.R, Cheng.M.J, Rosenberg.A.E, Mcgonegal.C.A, "A comparative performance study of several pitch detection algorithms", IEEE Trans. Accoustics, Speech, And Signal Processign, vol.ASSP-24, no.5, October 1976

[10] S.Shandilya and P.Rao, "Retrieving pitch of singingvoice from polyphonic audio", submitted to NCC-2003

[11] Doraisamy.S, "Locating recurring Themes in musical sequences", M.I.Ttech Thesis, University of Malaysia Sarawak, July 1995

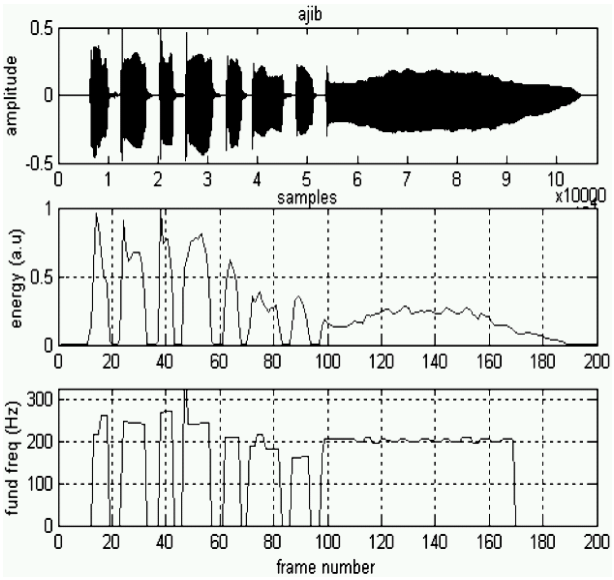

Figure 3. Waveform, energy contour and pitch track for the 8-note song phrase 'a-ji-b-daa-s-taan-he-ye' sung in syllable 'ta'.

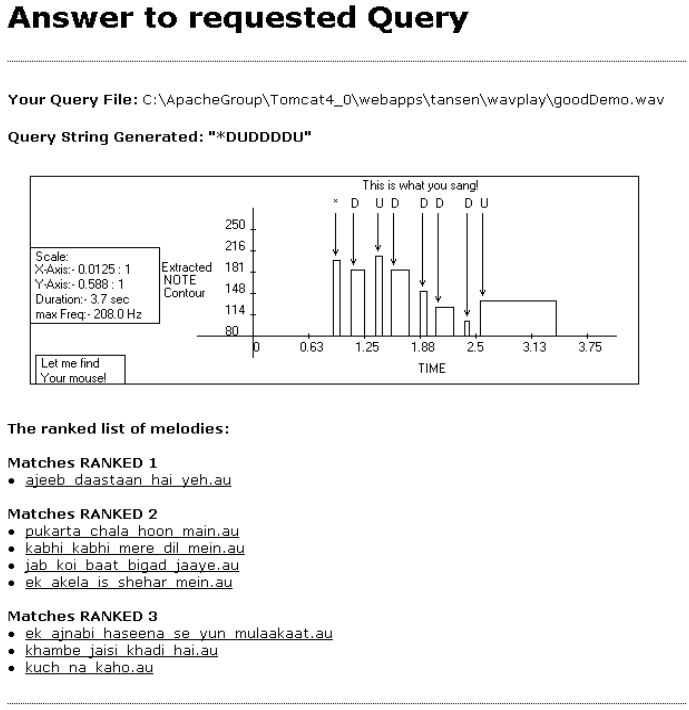

Figure 4. TANSEN user interface output screen in response to a query.