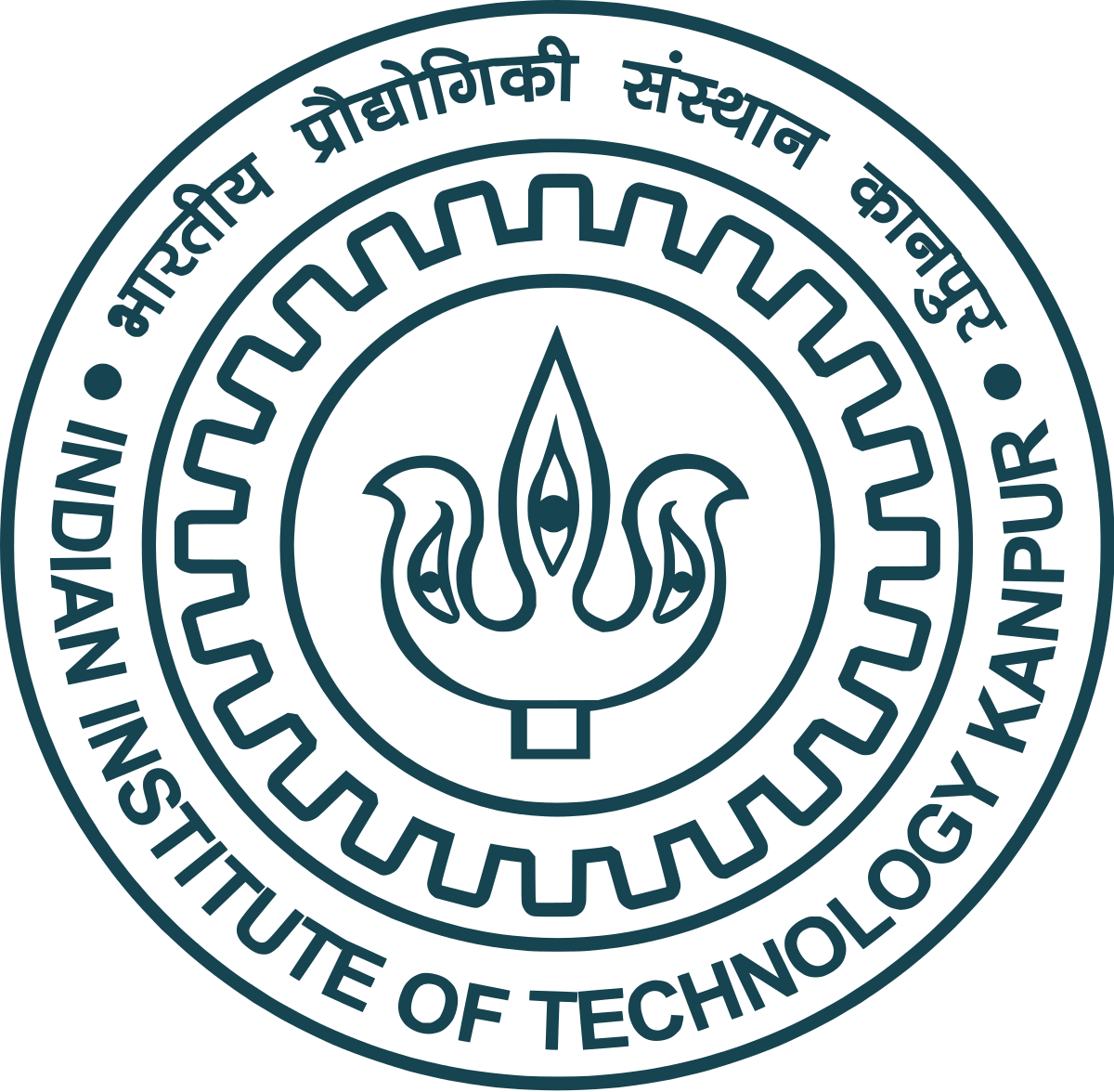
INDIAN INSTITUTE OF TECHNOLOGY KANPUR



PROJECT REPORT

TEAM 36

TOPIC: Query by Humming

-A Music Retrieval System

UNDER THE GUIDANCE OF: TEAM MEMBERS:

DR LAXMIDHAR BEHRA ENAKSHI

DR VIPUL ARORA RITIKA

DR HIMANSHU SINGH SATHWIK

Introduction & Objective:

A Query by Humming system allows the user to find a song by humming a part of the tune. Once the user hums into the microphone, the system records the hum and extracts certain features corresponding to the melody and rhythm characteristics, and it then compares the features to the features of the songs in the database. Finally, it returns a ranked list of the songs or song segments most similar to the humming. The main objective of the project is to design such a system using some of the learning techniques, and build a satisfactorily working prototype.

The QBH task can be broadly divided into two subproblems:

i) converting a query into a format which enables robust searching and

ii) matching the query with melodies in the database. The former problem is often associated with automatic transcription of a query into temporally segmented note events or into frame-wise measured pitch trajectory, whereas the latter concentrates on measuring melodic similarity.

Examples of QBH systems include ACRCloud, SoundHound, Musipedia, and Tunebot.

Problem Definition:

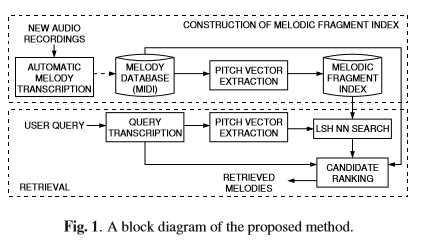
* Achieving a system with the capabilities defined above requires solving two parallel problems – the identification and evaluation of a parametric (or non-parametric) form for representing songs in the database and transformation of hums to the respective form, and the utilization of an effective learning technique to retrieve a ranked list of similar songs.
* As part of the first problem, parameters like pitch and spectral distribution of the hummed audio, or samples from song database, will need to be extracted, after an initial filtering. Using experimentation, an optimal representation of this data will be generated.
* The second part involves using this model and a chosen learning algorithm to train the system to retrieve matching songs from a database, based on what is learnt during training.
* Using machine learning algorithms, the training data would be a set of pre recorded hums for the various songs. This would form the data space. Decision boundaries would be drawn corresponding to

each of the songs. Whenever a user hums a new song, a new data point is created which is then classified into the class corresponding to its song based on the learning algorithm.

TYPES:

* Note based: The note-based methods extract and transcribe the note sequences from the hummed query, and then compare them against the note sequences of the main melodies in the database to retrieve the melody closest to the query. The humming transcription part often lowers the overall accuracy of the system, but they are much more efficient since comparing note sequences has significantly lower complexity than comparing the melodies frame by frame.
* Frame based: The frame-based methods use the extracted pitch to represent the melody and then use template-matching similarity measures such as DTW (Dynamic time warping) to measure similarity between the main melody and the query.

BLOCK DIAGRAM:



We propose a robust QBH method with sub linear search time over database items. Figure 1 shows a block diagram of the method. Given a database of melodies in MIDI format, the method constructs an index of melodic fragments by extracting pitch vectors. A pitch vector stores an approximate representation of melody contour within a fixed length time window. In retrieval, the method automatically converts a query into MIDI notes and then extracts pitch vectors. For each query pitch vector, the method searches for nearest neighbors in Euclidean space from the index of database melody fragments to obtain melody candidates and their matching positions in time. This can be performed very efficiently by using locality sensitive hashing (LSH). Final ranking of candidates is done by comparing the whole transcribed query to each candidate melody segment. Due to the melodic fragment index, the method manages long database melodies directly, without having to segment melodies into phrases. Also, the queries do not have to start from the beginning of a melodic phrase. Using LSH provides a signiﬁcant speed-up and retrieval performance comparable to the state-of-the-art.

To retrieve audio signals, we demonstrate the use of an automatic melody transcription method to produce melody database directly from music recordings and achieve very encouraging results.

ALGORITHMS:

* Locality Sensitive Hashing(LSH): It is an algorithmic technique that hashes similar input items into the same "buckets" with high probability.(The number of buckets are much smaller than the universe of possible input items.)Since similar items end up in the same buckets, this technique can be used for [data clustering](https://en.wikipedia.org/wiki/Cluster_analysis) and [nearest neighbor searc](https://en.wikipedia.org/wiki/Nearest_neighbor_search)h. It differs from [conventional hashing techniques](https://en.wikipedia.org/wiki/Hash_function) in that hash collisions are maximized, not minimized. Alternatively, the technique can be seen as a way to [reduce the dimensionality](https://en.wikipedia.org/wiki/Dimension_reduction) of high-dimensional data; high-dimensional input items can be reduced to low-dimensional versions while preserving relative distances between items.
* Dynamic Time Warping(DTW):It aims at aligning two sequences of feature vectors by warping the time axis iteratively until an optimal match (according to a suitable metrics) between the two sequences is found.
* Earth Mover’s Distance(EMD):Earth mover's distance (EMD) is a distance measure between two distributions, and has been widely used in multimedia information retrieval systems. Proposed the QBH system is made by combining the earth mover’s distance (EMD) and dynamic time warping (DTW) .

Constructing Index of Melodic Fragments:

The term melodic fragment refers here to a melody pitch contour within a ﬁxed length time window. The method constructs an index which stores melodic fragments, their temporal positions within the database melodies, and melody identiﬁers. The melody identiﬁer determines the song from which a melody fragment has been extracted. The index enables efﬁcient retrieval of melodies from the database.

FINAL RANKING:

We chose to use a note-based DTW measure similar to [20] with the note based EMD method. Then we use a parallel voting strategy to rank the available candidates, using the final scoring function.

File Formats:

* Wav File Format:

short form of the Wave Audio File Format. The most common use is to store an uncompressed audio. Quite large in size. First generation files of high quality

* MIDI File Format :

Musical Instrument Digital Interface. MIDI files are not exactly the same as the typical digital audio formats we use (like WAV, MP3, MP4 etc.). A MIDI is made up of information that describes what musical notes are to be played . MIDI Files therefore do not contain any 'real world' recordings

WAV TO MIDI CONVERSION :

To create a MIDI a file for a song recorded in WAV format a musician must determine pitch, velocity and duration of each note being played and record these parameters into a sequence of MIDI events. The Midi created represents the basic melody and chords of recognized music. The difference between WAV and MIDI formats consists in representation of sound and music. WAV format is digital recording of any sound (including speech) and MIDI format is principally sequence of notes (or MIDI events). Here we have an Output File (.mid) from an Input File (.wav) that contains musical data, and a Tone File (.wav) that consists of monotone data. An advantage of such a structure is also the fact that the query is prepared on the client side of the system. In this case the query is very short. Besides, there is a possibility to evaluate its quality before sending to the server. The system provides for playback of the recognized melody notes in MIDI format. This allows the user to listen to a query and take a decision either to send it to the server or to sing it once again.

\***This part is implemented using athem score software**

REFERENCES:

* <http://vlm1.uta.edu/~athitsos/publications/kotsifakos_petra2012.pdf>
* <https://www.ee.iitb.ac.in/web/files/publications/AnandBP2003.pdf>
* <http://www.slaney.org/malcolm/yahoo/Slaney2008-LSHTutorial.pdf>