PROJEKT QUEBEX:A QUERY BY EXAMPLE SYSTEM FOR AUDIO RETRIEVAL

Balaji Thoshkahna

Learning Systems and Multimedia Labs
Department of Electrical Engineering
Indian Institute of Science

K.R.Ramakrishnan

Learning Systems and Multimedia Labs
Department of Electrical Engineering
Indian Institute of Science

ABSTRACT

This paper describes an audio retrieval system,Quebex,that works on raw audio data. The system is able to retrieve songs that are rhythmically and timbrewise similar from a database of 700 songs. A new algorithm is proposed that attempts to match two songs temporally in terms of timbre,using a simple dynamic time warping algorithm. A sieving technique is used to eliminate unlikely candidates from further processing and a ranking system among the rhythm, spectral and temporal features is used to retrieve songs. Initial results have shown great promise for the approach as it retrieves remixes and dubbed songs within the top 10 retrievals.

1. INTRODUCTION

Music information retrieval(MIR) is a fast growing field with vast potential.A Google like search engine for audio will be of great help for music fans and will be of immense use to repositories like Napster etc.With the growth of popularity for Indian films around the globe, the popularity of Indian film music also has grown exponentially-the World Wide Web contributing hugely to this.There are professionally maintained websites that cater to Indian music fans (www.raaga.com,www.indiamusiconline.com etc). But the search engines in those sites are extremely simple and text based.Also searching for similar songs is not possible.

2. PREVIOUS WORK

We describe a Query by Example(QBE) system that takes a piece of music as input and retrieves "similar" music pieces. Similarity is an ill defined property,especially when it comes to a complex signal like polyphonic music.Cuidado system[1] defines similarity as having similar instrumentation("global timbre") in the query piece of music and the retrieved music pieces.Foote[2] attempts audio retrieval by defining "rhythmic similarity" by comparing two songs spectrally. Logan and Salomon[3] describe a system that tries to identify similarity based on Earth Mover's Distance between two pieces of music whose frame based features have been k-means clustered.Cheng[4] considers retrieval based on spectral similarity.Spevak et.al[5] describe a system (Soundspotter) that attempts to match the MFCC trajectory of selected sounds in an audio piece by using a DTW based algorithm.The similarity functions used in the above examples is usually spectral content based,with little or no regard to the tempo(perceived speed of a performance) features.One way to account for tempo would be to use average tempo[6] as a feature, and then consider the retrieved song as being similar only if its tempo also matches,along with spectral features.But here also the background percussion track(drums,tabla,hi-hat etc) can vary between the two retrieved songs and hence the perception of similarity can become bad.Also,similarity is calculated using various distance measures,but in all cases there is some form of subjective evaluation done on a set of music pieces[1, 3, 4]

3. QUEBEX ALGORITHM

Our algorithm uses an approach that combines the techniques used in [1, 2, 5, 6].We use tempo information along with timbre information to try and define a stricter form of similarity,than those defined above.Two audio pieces are considered "similar" by us if they are:
1.Rhythmically similar(Have same tempo and similar percussion track('taala')[1])
2.Temporally have similar timbre(Ex:Lets say song A has violin followed by piano,we say song B is similar to song A if it has a section where violin is played followed by piano.We like to match the timbre's of the two songs temporally also.)
Condition 2 above retrieved extremely good matches when there were enough number of similar songs in the database.

The following sections explain our algorithm in detail.

[1] 'taala' is the Indian music term equivalent to 'meter' or style of percussion performance

3.1. Database Information

The database has 700 music pieces each of around 10 seconds(average length) taken from CD recordings and converted to '.wav' fi lesat a sampling rate of 16kHz.The database contains songs from various Indian regional language movies and albums of various international artistes and has various genres of music(though not labelled by us).

3.2. Extraction of Spectral Feature set

Each song is divided into non-overlapping frames of 20 milliseconds.Features such as mean and standard deviation of spectral flux($Flux_{avg}$ and $Flux_{std}$) and the number of zero crossings (ZCR_{avg} and ZCR_{std}) are computed [7].

13 point MFCCs(Mel frequency cepstral coeffi cients) are computed for each frame and the fi rst8 coeffi cientsare used to represent the timbre of the song[1, 8]($MFCC(i,j)$ is the MFCC vector for the *jth* frame of *ith* song).Since we have 10 seconds of music on an average this comes to around 500 frames.

We now split the 10 sec music piece into frames of non-overlapping 0.5 seconds.We take the mean and standard deviation of the MFCCs(20 msec frames) falling within each 0.5 second window and represent the group of frames within the 0.5 second window by the above mean and standard deviation vectors.

$$MFCC_{Avg}(i,j) = \sum_{k=(j-1)*25}^{j*25} MFCC(i,k) \quad (1)$$

$$MFCC_{Dvn}(i,k,j) = ||MFCC(i,k) - MFCC_{Avg}(i,j)||^2. \quad (2)$$

$$MFCC_{Std}(i,j) = \sqrt{\sum_{k=(j-1)*25}^{j*25} MFCC_{Dvn}(i,k,j)} \quad (3)$$

where j varies from 1 to 20.

This way we have a trajectory of MFCCs of 40 vectors(20 for the mean vectors and 20 for the standard deviation vectors),for every song.

3.3. Extraction of Temporal and Energy features

For each music piece,the percentage of frames that have energy less than the average energy of the music piece (Per_Eavg) is calculated and stored.The mean and standard deviation of the song energy across frames is computed ($Song\_Eavg$ and $Song\_Estd$).

3.4. Extraction of rhythmic features

An image processing approach based on [9]was used to extract the percussion track in each song.We assume,in our algorithm that each song has some percussive signal which is very valid since most Indian fi lmsongs have percussion tracks(mridangam,tabla,ghatam,drums etc).Also the database has over 500 of Indian songs.The mean and standard deviation of energy in the percussion signal is calculated.Energy of the percussion signal is assumed to be the energy of the frame that is declared to be a percussive frame($PercEn\_avg$ and $PercEn\_std$).Once that is extracted,a smoothed version of the percussive track is used, and its autocorrelation taken and normalized and downsampled to a 50-point template.($Templ\_acf(i)$ for the *ith* song) (Note in the Fig1 that the autocorrelation function of Song 3 is different from that of Song 2 and Song 1,since Song 1 and Song 2 are dubbed versions of the same song.)The number of percussion hits per second is taken as the average tempo of the song($Tempo\_avg$) (An assumption that goes bad for a few songs when the percussion track is not maintaining the tempo). The average and standard deviation of percussion strength is a good indication of how strong the percussion signal is and how consistent it is.

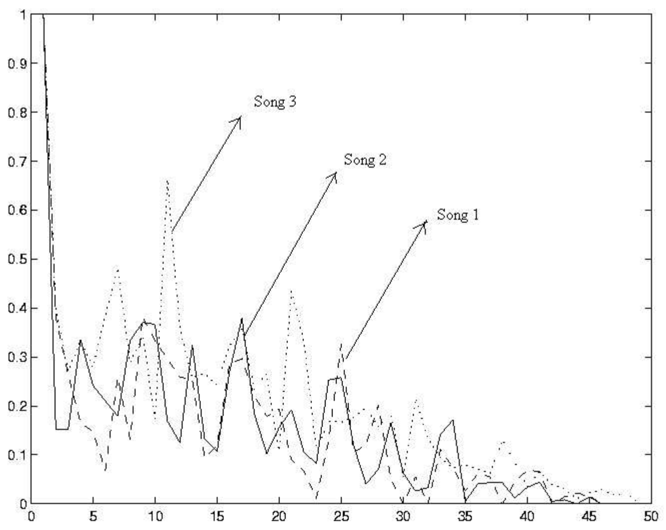

Fig. 1. Rhythm feature based on normalized autocorrelation template.Song1 and Song2 have similar autocorrelation functions as compared to Song3.

4. ALGORITHM FOR RETRIEVAL

A query song can be selected from the database itself using the MATLAB GUI(Fig.3).A song can be played(Using "Play song" button) before selecting(Using "Select song" button)it as the query song.The algorithm will search for similar songs in the database and retrieve the nearest 10 songs based on a ranking system.

The following steps were used for retrieval purposes: **(1).**Given a query song A,a symmetric distance between the MFCC vector tracjectories of song A and all the songs in the database,was calculated as shown in Fig.2.That is,in the 8 dimensional MFCC space,we try to map each point of query song A to the nearest point in song B's trajectory (with some temporal restriction(Fig2)).This allows for a temporal matching of the timbres between the two songs without restricting too much and

$$d(A,B) = \sum_{i=2}^{19} min\{||(A(i)-B(i-1)||^2, ||A(i)-B(i)||^2, ||A(i)-B(i+1)||^2\} + min\{||A(1)-B(1)||^2, ||A(1)-B(2)||^2, ||A(1)-B(3)||^2\} + min\{||A(20)-B(18)||^2, ||A(20)-B(19)||^2, ||A(20)-B(20)||^2\} \quad (4)$$

where A(i) is the ith MFCC vector in the trajectory of song A and d(A,B) is the distance of song A wrt song B.(The last two terms of the above equation is to take care of the starting and end points)

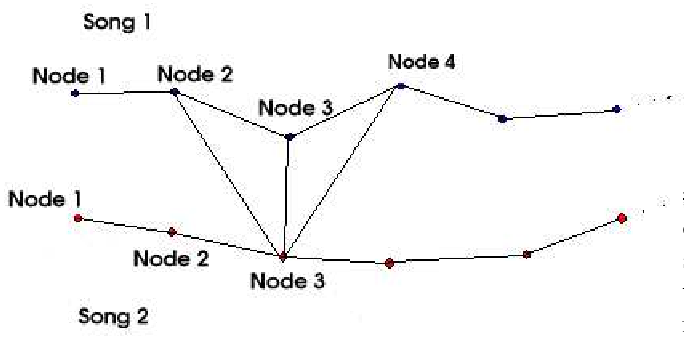

Fig. 2. A suboptimal DTW mapping for distance calculation.Each node in the fi gurecorresponds to a 0.5 sec window's mean/standard deviation vector in the 8 dimensional space.

We make the distance measure symmetric,by taking an average between the two mutual distances;

$$D(A,B) = (d(A,B) + d(B,A))/2. \quad (5)$$

The above distance is used on two features namely,the $MFCC_{Avg}$ and $MFCC_{Std}$.This distance measure is intuitively satisfying enough,though we can consider more complex and optimal DTW mappings to fi ndthe distance

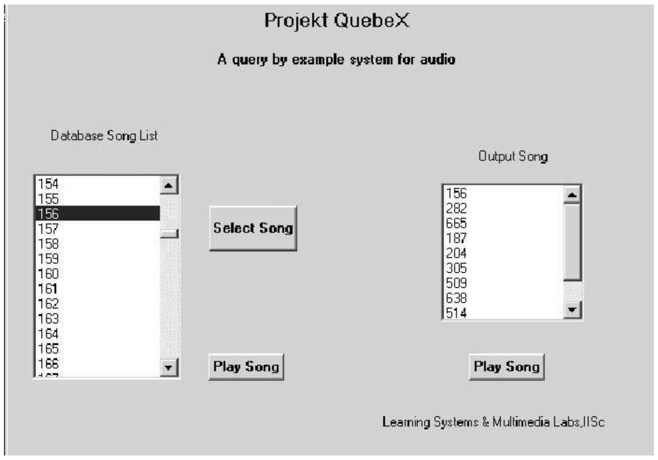

Fig. 3. Projekt Quebex's MATLAB GUI

between the two trajectories.The songs are ranked according to their distance from the query song($R1$).Now,Euclidean distance between the features $Flux_{avg}$, $Flux_{std}$,ZCR_{avg} and ZCR_{std} of the query song A and a database song B is calculated as follows;

$$d_{favg}(A,B) = (Flux_{avg}(A) - Flux_{avg}(B))^2. \quad (6)$$
$$d_{fstd}(A,B) = (Flux_{std}(A) - Flux_{std}(B))^2. \quad (7)$$
$$D\_Flux(A,B) = d_{favg}(A,B) + d_{fstd}(A,B). \quad (8)$$
$$d_{zavg}(A,B) = (ZCR_{avg}(A) - ZCR_{avg}(B))^2. \quad (9)$$
$$d_{zstd}(A,B) = (ZCR_{std}(A) - ZCR_{std}(B))^2. \quad (10)$$
$$D\_Zcr(A,B) = d_{zavg}(A,B) + d_{zstd}(A,B). \quad (11)$$

for all songs in the database.The distances are again ranked ($R2$ for D_Flux and $R3$ for D_Zcr).Ranking is intuitively satisfying since the features have different orders of magnitude and adding them up as in Euclidean distances can mean neglecting the importance of some features. A fi nalrank R is calculated as,

$$R = R1 + \alpha.R2 + \beta.R3. \quad (12)$$

where α and β are constants of weighing the features $R2$ and $R3$ respectively.(This means that different features can be given different importances.We gave a value of 1 to α and 0.25 to β,meaning that the zero-crossings measure which relates to *'noisyness'* in a music piece was neglected as compared to the spectral distance measures $R1$ and $R2$)

(2). All the songs ranked in the top 50% of the above rank R,are returned for further processing.This sieving eliminates songs whose timbre and spectral features are way too different from the query song.

(3). Now a Euclidean distance is calculated between the temporal features(*Per_Eavg*,*Song_Eavg* and *Song_Estd*) of query song and the database songs and the songs are ranked.

These ranks are clubbed together and a similar ranking ("temporal feature matching rank"),as explained above is calculated.($R\_temporal$)

(4).Euclidean distance between the autocorrelation templates(Calculated as the sum of the differences between the corresponding points of each template),the mean tempo and percussion strengths(*PercEn_avg*,*PercEn_std*,*Templ_acf(i)* and *Tempo_avg*) are computed between the query song and the database songs and a weighted ranking(Each of the above said features are ranked and then the ranks are added based on weights.Ex:If we want to match the percussion autocorrelation templates better,greater weightage is given to that ranking)is used to get a "rhythm feature matching rank" (R_rhythm)

(5).Now the 2 ranks are added to get a finalrank for each of the song.

$$R\_final = R\_rhythm + R\_temporal. \qquad (13)$$

(6).The top 10 ranked(R_final) songs are retrieved.The query song will always be ranked 1.This ranking system resulted in a better retrieval than simple Euclidean distance between the features.
We believe that the human brain attempts to look for the "strongest" feature in a song and finallytakes a decision of similarity based on the perceptual strength of features(Ex: Users of our system were able to accept 2 rock songs that had different instrumentation,tempo and rhythm, but same "noisyness"(related to zero crossing rate) as "similar" songs). But we don't have a way to deduce which feature dominates over the others.This makes retrieval by similarity challenging.Preliminary tests in our labs indicate that our direction of thought is promising.

5. RESULTS

10 songs were randomly chosen from the database and their retrievals were tested on 5 people.For these 10 songs,users reported an average of 60% of the songs to be similar to the query songs.Also the system always retrieved remixes, dubbed versions of the same songs well within the top 10 songs.This was a self-test,in which,the system worked well. The retrieval was bad when there were not enough songs of the type of the query song.This has motivated us to increase the database size to around 1500 or more songs.([1, 3] had huge databases to start with and therefore must not have felt this problem.[2, 4] had very few songs(around 200) and must have had this problem(We tested the outputs for system by Foote[2] and felt this problem for some songs).

6. CONCLUSIONS AND FUTURE WORK

QBE systems that are based on perceptual similarity measures appear to be promising. Appropriate characterization of a song/piece of recorded music ,such as done by listeners, in terms of *bass,noise,rhythm etc.* may be incorporated into QBE systems. Finding suitable computational features and descriptors to match such a characterization is a challenging task. We may need to use more than one feature set for this purpose since no single feature might contain all the information for such a characterization. Feature augmentation and clustering, sieving out the principal component features for pattern matching are interesting problems for future research.

7. ACKNOWLEDGMENTS

We would like to thank Krishna.A.G,Chandra Sekhar.S and all our lab members for the help and encouragement.

8. REFERENCES

[1] J.J.Aucouturier and F.Pachet, "Music similarity measures:what's the use?," *Proceedings of the 3rd International Symposium on Music Information Retrieval*, 2002.

[2] M. J.Foote and U.Nam, "Audio retrieval by rhythmic similarity," *International Symposium on Music Information Retrieval*, 2002.

[3] B.Logan and A.Salomon, "A content based music similarity function," *Cambridge Research Labs - Tech Report*, June 2001.

[4] C. Yang, "Macs:music audio characteristic sequence indexing for similarity retrieval," *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics* ., 2001.

[5] C.Spevak and E.Favreau, "Soundspotter-a prototype system for content based audio retrieval," *5th Intl. Conference on Digital Audio Effects(DAFx-02)*, 2002.

[6] E.Scheirer, "Music listening systems," *PhD Thesis,MIT*, 2000.

[7] G.Tzanetakis and P.Cook, "Marsyas:a framework for audio analysis," *Organised Sound*, 2000.

[8] B. Logan, "Mel frequency cepstral coefficientsfor music modeling," *International Symposium on Music Information Retrieval*, 2000.

[9] A.Klapuri, "Sound onset detection by applying psychoacoustic knowledge," *ICASSP*, 1999.