Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 8, August 2015, pg.216 – 225

RESEARCH ARTICLE

Music Information Retrieval Using Hit Count Vector

Yasmin A. Hassan[1], Loay E.George[2]
Dept. of Computer Science, College of Science, Baghdad University, Baghdad, Iraq
[1] yasmin_alaa85@yahoo.com, [2] loayedwar57@yahoo.com

Abstract: Roughly speaking, music can be easily and cheaply accessed via Internet, but at the same time, finding certain music is still a difficult task. In this paper, a music retrieval system based on frequency descriptors is presented. The work flow of proposed system passes through two main phases: (i) the enrollment phase and (ii) the retrieval phase. In both phases, two models are presented (i.e., preprocessing and feature extraction module). Transformation is an important stage in audio processing application; it is applied to convert the audio signal from time domain to frequency domain. Two types of frequency transformation have been used in this work: (i) Fourier and (ii) discrete cosine transform (DCT). The proposed method implies the application of several processing stages on the input melody files; these stages are considered as parts of the preparation operation. The task of music features extraction for retrieval purpose is a challenging problem. An optimization algorithm is developed to determine the proper threshold value that can filter-in a set of the most energetic and frequent frequencies existing in the spectra, such that their number should be sufficient. This set of frequencies is used to establish the discriminating features, called *hit count* array.In the enrollment phase, the extracted hit count array is stored in the system database as a template to be used in the retrieval process phase. Euclidean and City Block similarity metrics have been used to make a decision in the retrieval stage. For performance evaluate, several performance measures (like, sensitivity, specificity, recall, precision, accuracy) have been used.

The system was tested using a dataset consists of 50 audio files; each has the specifications (11025 sampling rate, 8-bit resolution and mono channel). The length of taken Melodies is about one minute; they saved in a database. Each melody sample is partitioned into 30 frames; each frame is about two seconds. The achieved retrieval results indicated high recognition rate (99.45%) when using the input query length equal to 35 seconds.

Keywords:Fourier, DCT, Frequency Distribution Descriptors,Spectra Matching, Distance Measures.

1. Introduction

Music is very popular in modern life, and the quantity of digital music available to music listeners has increased considerably. In computer science, researchers have been intensely working on developing techniques for dealing with digital music data. In particular, the development of efficient and effective computational assistants in music listening has recently become more and more vital due to the high demand for web-based music stores and services. An important subject in computer-aided music listening is music retrieval. (i.e., the problem of efficiently, helping users in locating the music they are looking for) [1].

Due to the diversity and richness of music, MIR research brings together experts from a multitude of research fields ranging from information science, audio engineering, computer science, musicology, music theory, library science to law and business [2].

Bandera (2011) proposed a system that uses a database with real songs and does not need another type of symbolic representation of them. The system employs an original fingerprint based on chroma vectors to characterize the humming and the reference songs [3]. 2D Fourier transform have been used for large scale recognition tasks by Therry and Daniel (2012). They indicated that simple Euclidean measures are unsuitable for rendering tasks [4]. Urbano, Schedl and Serra (2013) have made suggested several ideas to carry out evaluations relevant to the field of Music Information Retrieval. The investigated evaluations depend on previous issues discussed in the context of Text IR field from the point of view of validity, efficiency and reliability of the experiments. Several issues have been studied in terms of evaluation for the purpose of improvement in [5].

2. Music Retrieval Using DCT and Fourier Techniques

In this research a retrieval system is proposed its input is a short excerpt without any other information and keywords, and its output is an audio wave file that the short excerpt belongs to. The data of the input melody is a wave file (in time domain). It is transformed from time domain to frequency domain. In our proposed system we used two transforms (i.e.; DCT and Fourier), and from the transforms coefficients a set of discriminating features is extracted to be used for distinguishing the melodies saved in the system database.

2.1 The Proposed System Layout

The work flow of the proposed system passes through two main phases: (i) enrollment phase and (ii) retrieval phase. In both phases, the system starts with same first two stages (i.e.; preprocessing and feature extraction).

The preprocessing stage consists of:

A. Load and prepare the melody wave file.
B. Apply normalization on the wave samples such that their values to be within the range [-1,1].
C. Partition the wave data into blocks.
D. Apply second normalization for wave array (i.e., normalize all melody signals to have the same power energy).
E. Transform each block data to the frequency domain (either Fourier or DCT).

Feature extraction stage is applied after the preprocessing stage. The feature extraction module consists of:

A. Determine the root mean square of each frame of audio wave file.
B. Applying threshold optimization algorithm to select a proper threshold that is suitable with each frame.
C. Apply thresholding and statistical analysis to allocate the energetic and most frequent frequencies.

In the enrollment phase, after preprocessing and feature extraction step, the extracted features vectors, extracted from all melody wave file, are manipulated to determine the templates. Then, these templates are saved in the feature database. While in the retrieval phase, the extracted features vector is matched with the templates stored in the database to make a decision.

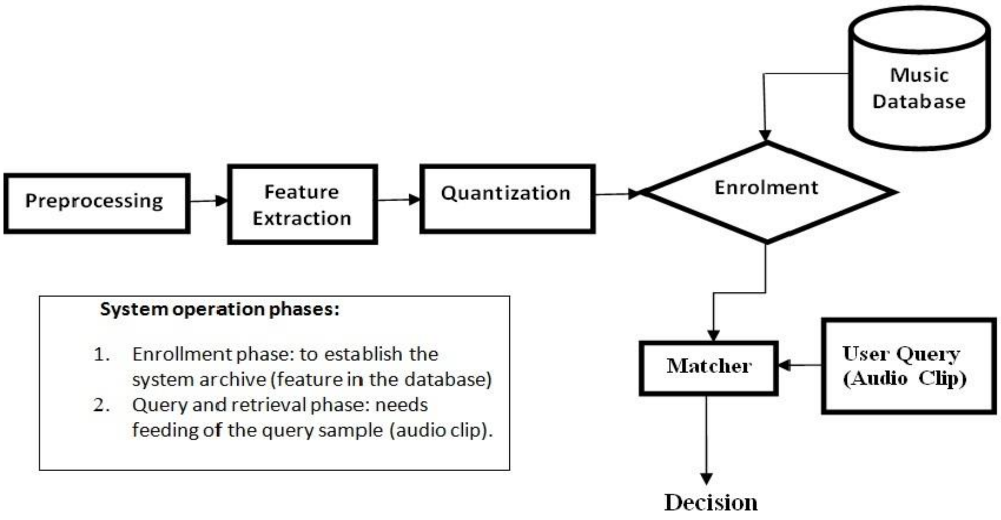
Music Database
Preprocessing
Feature Extraction
Quantization
Enrolment
System operation phases:
1. Enrollment phase: to establish the system archive (feature in the database)
2. Query and retrieval phase: needs feeding of the query sample (audio clip).
Matcher
User Query (Audio Clip)
Decision

Fig.(1): The Layout of the Proposed Retrieval System

2.2 Audio Transformation

The purpose of transform to frequency domain is to convert data to a form that can be more descriptive for melody data. In this stage the audio signal is transformed from time domain to the frequency domain; this can be achieved by applying Fourier transform or discrete cosine transform (DCT). These two transforms have been applied as alternative transform options.
The DCT in one dimension is given by [6]:

$$G(f) = \sqrt{\tfrac{2}{N}}C_f \sum_{t=0}^{N-1} p_t \cos\left(\frac{\pi f(2t+1)}{2N}\right) \qquad (1)$$

The discrete Fourier transform is [6]:

$$G(f) = \sum_{t=0}^{N-1} p_t \left[\cos\left(\frac{2\pi f t}{N}\right) - \mathrm{i}\, sin\left(\frac{2\pi f t}{N}\right)\right] \qquad (2)$$

2.3 Features Extraction and Evaluation

After conducting the preprocessing stage, features extraction module is applied on the transform spectra (whether it is Fourier or DCT spectra) in order to make successful matching between music parts. In this module, a set of high valued and most frequent frequencies are assembled as features vectors, which is stored in a dedicated database, this feature vector is considered as a template representing the melody. This module has three stages; the first one is the determination of root mean square for the spectra of each frame; and to take this value as an initial threshold value in the next stage. The second stage is implies the application of an optimization algorithm to select a suitable threshold value. The third stage includes the determination of the histogram for the high value peaks in the frequency based spectra. The set of most redundant strong frequencies is assembled as a template; called "***hit count array***". This module is the kernel step for the suggested retrieval (or recognition) system. Figure (2)presents the main steps of the feature extraction selection module.

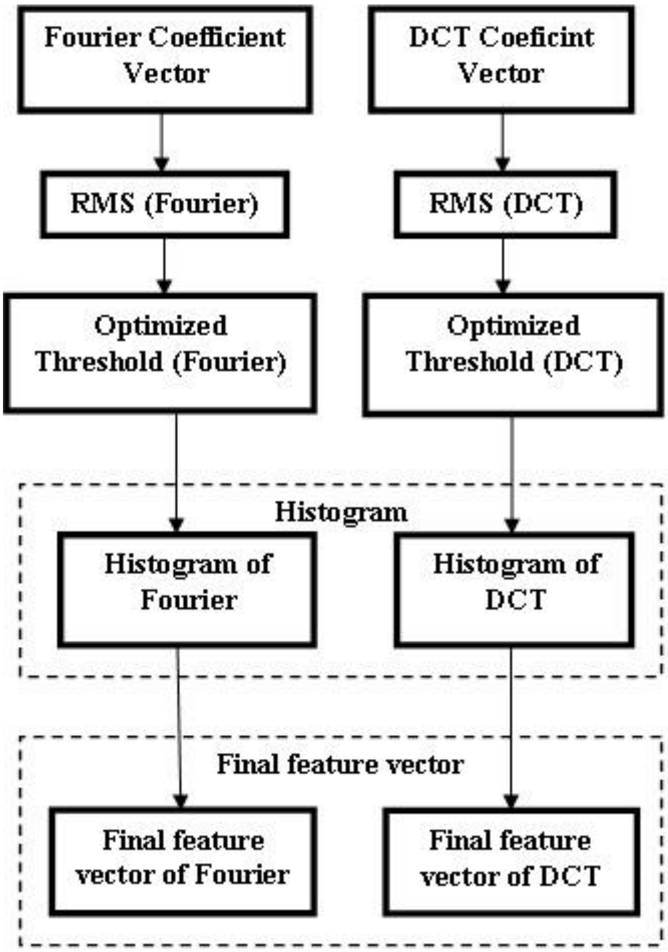
Fourier Coefficient Vector
DCT Coeficint Vector
RMS (Fourier)
RMS (DCT)
Optimized Threshold (Fourier)
Optimized Threshold (DCT)
Histogram
Histogram of Fourier
Histogram of DCT
Final feature vector
Final feature vector of Fourier
Final feature vector of DCT

Fig.(2) Block diagram of feature extraction and evaluation module

2.4 Determination of Optimized Threshold

The first step in this stage is computation of root mean square to be the initial threshold value in the optimization algorithm; which is introduced to select the suitable threshold for each frame.Therootmeansquareisdeterminedusingthe following equation:

$$RMS = \sqrt{\sum \left(w(i)\right)^2} \qquad (3)$$

The second step in this stage is the computation of suitable threshold for each frame. The input to the algorithm is the RMS (root mean square) as initial value for the algorithm and the second input is either DCT or Fourier coefficients array of the tested frame. The output is a suitable threshold needed to extract the energetic frequencies whose coefficients values are more than the threshold value. The adopted proper threshold is that can cause auto selection of [120-140] number of energetic frequencies.

Initially, the predetermined RMS is taken as the initial threshold value. Then, if it satisfies the condition of needed range of frequencies, then it will be considered as proper threshold value. Otherwise, the threshold value is incremented one step size (i.e.; 0.1). This step size is taken as the change in threshold value; this change may be positive or negative depending of the current attained frequencies, when this number is less than the lower bound (i.e., 120) of required frequencies then the threshold change is taken positive; while when the number of counted frequencies is higher than the upper bound (i.e., 140) then frequency change value is taken negative.

Since using MSE as initial threshold value, then at first the number of counted frequencies (i.e., whose values more than the threshold) is higher than the upper bound. Then a loop of gradual increase is made until reaching the optimized threshold value that can lead to the suitable number of frequencies. In case the determined frequencies become less than (or even changed to be larger than) the range [120-140]; then the step size will be divided by 3, in such case the change step will become smaller to loop keep trying till getting the acceptable number of energetic frequencies. Algorithm (1) presents the threshold optimization steps.

```
Algorithm (1) Optimized Threshold
Input: RMS //root mean square,
        DCT frame or Fourier Frame
Output:optimized threshold
Procedure:
Step 1: //Set initial value
        Set thr=RMS
Step 2:// determine step size
St_size=0.1*thr
Step 3:// calculate no. of values more than or equal the determined threshold
Set c=0
        For all i Do {where 0< i<frame size}
  If (f(i)>= thr) then increment c
        End For
Step 4:// determine range
        Set n1=120,   n2=140
Step 5:// test
        If (c<= n1) and (c<=n2) then Set  Opt_th=thr,   Opt_N=c
Step 6:// determine the direction
        If (c<n1) thenSetsign=1Else Set sign=-1
Step 7:// determine the initial value of flage
        Set flag=false
Step 8:// loop
       Do loop
thr=thr+sign*st_size
Step 9:// repeat step //calculate no. of values more than or equal the determined threshold
Set c=0
For all i Do {where 0< i<frame size}
If (f(i)>= thr) then increment c
           End For
Step 10:// test
 If (c<n1) and (sign=1) then
SetSign=-1:   St_size=St_size/3
 Else If (c>n2) and (sign=-1) Then
Set Sign=1:  St_size=St_size/3
Else If ((c>=n1) and (c<=n2))
Flag=true
           End If
Step 11://
If (thr<= 0) then thr  = 0.1
Do Loop While (flag=false)
Step 12: Return thr
Step 13: End.
```

2.5 Applying Histogram

At this stage, most intensive and frequent frequencies are selected as attributes needed to make the comparison step. The input to this stage is a sub-set of Fourier coefficient vector or DCT coefficient vector. The histogram is computed to each transformed spectra (i.e., one of two histograms is determined; either for Fourier or for DCT).

In the histogram computation stage, a loop is done on the input block array to compare each element in the array with the optimized threshold. If the result of comparison is more than the threshold value, then the content of the element histogram array is incremented by one. When this scan loop is applied overall the blocks of the input melody data, then the histogram is considered as ***the hit count array***. This array refers to the number of high and frequent frequencies appeared in all blocks. Note that the size of histogram array is equal to the same size of block size.

```
Algorithm: Applying Histogram
Input: block array of Fourier or block array of DCT coefficient,threshold
Output: histogram array  (Hit count array)

Procedure:
Step 1://loop for all input block array
          For all i Do { where 0 < i < block length }
Step 2://compare each element of block array with threshold
    If (ff(i) > th) then his [i]=his[i]+1
         End For
Step3:Return  histogram array
Step 4: End.
```

2.6 Retrieval Method

This module requires a query audio sample (i.e., an excerpt of music) which should fed by a user, to be matched with the records registered in the existing database to make final decision by the system. The query is a short musical excerpt (i., has record time about 30-35 seconds) without any other information and keywords. All processes that are applied on the original wave audio file will be applied on each input query. These processes implies preprocessing operations (i.e., preparation, partitioning, normalization, applying Fourier and DCT) and feature extraction (i.e., RMS, optimized threshold, histogram) to get the descriptive features of the input small query.

Then, comparison is made by applying one of the standard distance measures; like the mean absolute difference (MAD):

$$MAD = \sum_{i=0} \left| S_k(i) - T_j(i) \right|$$

Where S_k is the extracted hit count array of the k^{th} melody sample, $T_j$is the hit count array of j^{th} template stored in the system database. The melody template that shows nearest distance is specified as the most nominated melody to which the short excerpt belongs. Algorithm (2) presents the implemented steps of the retrieval decision stage.

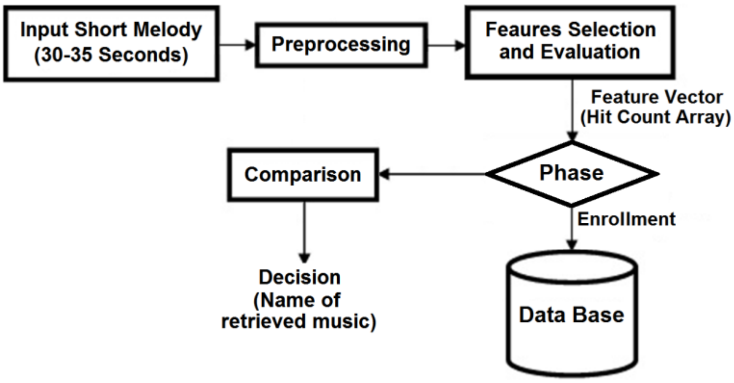
Input Short Melody (30-35 Seconds)
Preprocessing
Feaures Selection and Evaluation
Feature Vector (Hit Count Array)
Phase
Comparison
Enrollment
Decision (Name of retrieved music)
Data Base

Fig. (3) Block diagram of Retrieval Method

Algorithm (2): Music Retrieval using hit count vector

```
Input: Short music excerpt may be 30 second.
Output: Original melody that short music excerpt belongs to.
Procedure
Step 1: // Prepare the input short melody wave and convert it to array of values.
Step 2: //First Normalization.
Step 3: //Partitioning the wave according to determined parameters like time, sample rate, melody
      length, block size, as done for the original audio wave files.
Step 4: //Another normalization according to equation
Step 5: //Applying DCT and Fourier coding techniques.
Step 6: //Extract RMS for DCT and Fourier.
Step 7: //Computeoptimized threshold for DCT and Fourier.
Step 8://Apply histogram for DCT and Fourier to determine hit count array as needed features.
Step 9://Compare this hit count with the whole saved database by using MAD distance measure.
Step 10:// Find the smallest or nearest MAD and determine the name of that melody which the input short melody belongs to.
Step 11: End.
```

3. Test Results

In this work, 50 samples of audio wave files have been used for training and testing purpose. The taken samples include the following properties:

1. They have wave file format (.wav),
2. Have sample resolution= 8 bit/sample,
3. The sampling rate is 11025 sample/sec,
4. With mono channel,
5. The length of each audio wave file (in time) is 1 minute,
6. Its data size is 664 KB.

Each melody files had been partitioned into 30 frames; with frame duration equal 2 seconds. The type of all taken melody audio files for testing purpose is "soft music" (authored by Yanni, Giovanni, Bithawvin).

Six melody samples have been taken as audio testing material to study the performance behavior of the system. Figures (4-9) present the original shape of the taken melodies as testing samples.

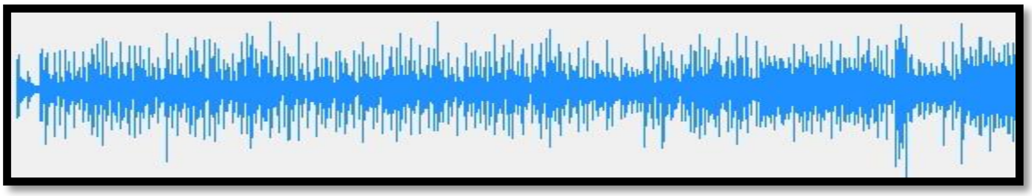

Fig. (4) Original Shape of test1.wav

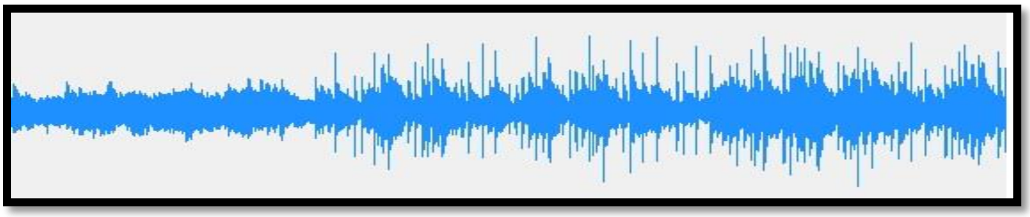

Fig. (5) Original Shape of test2.wav

Fig. (6) Original Shape of test3.wav

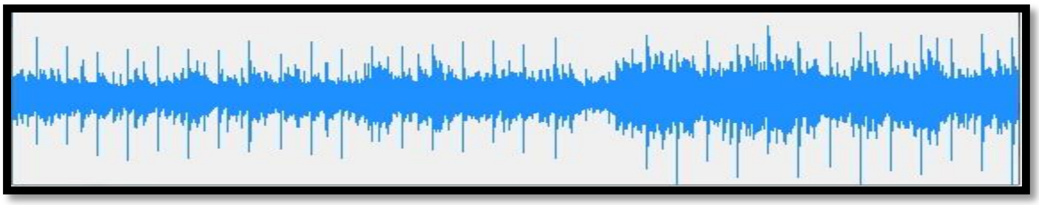

Fig. (7) Original Shape of test4.wav

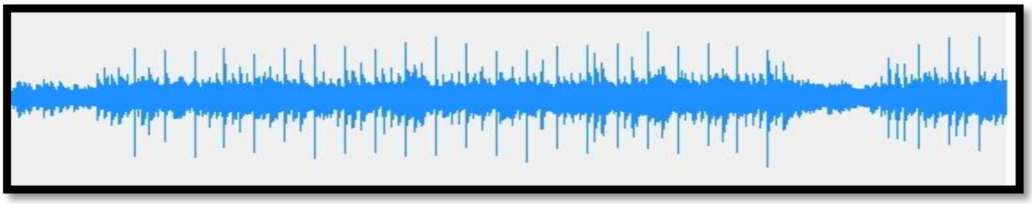

Fig. (8) Original Shape of test5.wav

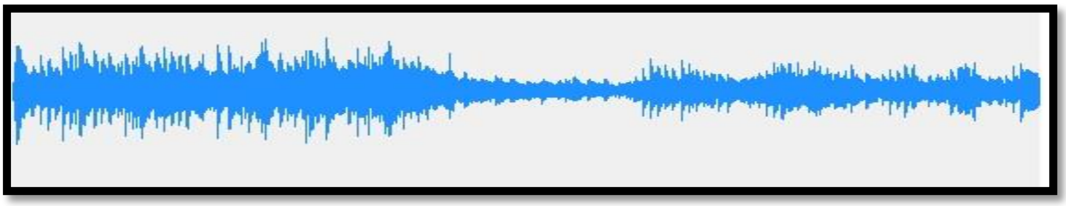

Fig. (9) Original Shape of test6.wav

3.1 System Performance

Through the conducted tests, it was noticed that the system had mostly succeeded in retrieving the most relevant responses (i.e., the original melody that the input query belongs to). Several retrieval performance measures (like, TPR, TNR, PPV, NPV, FPR, FDR, FNR, ACC, F1 score, MCC, Informednes, Markedness) have been determined for describing the retrieval performance.The performance measures depend on the confusion matrix which is computed for the whole test classes.

The system was tested using six melodies classes. When the default system parameters (i.e., 1 minute melody blocks partitioned into 30 blocks non-overlapped blocks) were used, the attained average retrieval accuracy (ACC) was found 99.45%.

3.2 Duration of Input Query

Several tests have been done for the duration of input query. The system retrieves the original melody from the database when the query has length in the range [30-35] second; if the length is less than this range [30-35] it was noticed that the system doesn't always produce correct retrieval decision.

The retrieval process depends on the standard distance measures such as the mean absolute difference (MAD) and the mean square difference (MSD). When the input query has length close to (1) minute the values of MAD and MSD will greatly reduced.

The duration of input query must be sufficient to make good capturing for the exact features of the audio signal. If the length of input query is less than 30 seconds, then the system faces failure in retrieving the music file correctly. But, when the input query length is 20-25 seconds, then the results will be acceptable when Fourier transform is used, and it is significantly failed when DCT transform is used.

The following tables (1 to 4) illustrate the effect of input query duration on the retrieving process.

Table (1) The MAD and MSD Values for Both DCT and Fourier Based Methods, for Input Query test3.wav (its duration is 30 second)

Melody Name	DCT		Fourier	
	MAD	MSD	MAD	MSD
test1.wav	0.172	0.307	0.172	0.345
test2.wav	0.170	0.285	0.164	0.384
test3.wav	0.083	0.136	0.086	0.301
test4.wav	0.171	0.291	0.170	0.391
test5.wav	0.168	0.289	0.175	0.417
test6.wav	0.167	0.274	0.169	0.431

Table (2) The MAD and MSD Values for Both DCT and Fourier Based Methods, for Input Query test5.wav (its duration is 30 second)

Melody Name	DCT		Fourier	
	MAD	MSD	MAD	MSD
test1.wav	0.174	0.344	0.172	0.310
test2.wav	0.166	0.394	0.164	0.279
test3.wav	0.172	0.435	0.171	0.306
test4.wav	0.172	0.399	0.166	0.277
test5.wav	0.088	0.304	0.083	0.132
test6.wav	0.17	0.431	0.161	0.259

Table (3) The MAD and MSD Values for Both DCT and Fourier Based Methods, for Input Query test2.wav (its duration is 35 second)

Melody Name	DCT		Fourier	
	MAD	MSD	MAD	MSD
test1.wav	0.172	0.298	0.175	0.309
test2.wav	0.069	0.200	0.067	0.101
test3.wav	0.169	0.352	0.178	0.313
test4.wav	0.168	0.327	0.171	0.282
test5.wav	0.171	0.339	0.170	0.287
test6.wav	0.167	0.355	0.165	0.260

Table (4) The MAD and MSD Values for Both DCT and Fourier Based Methods, for Input Query test5.wav (its duration is 35 second)

Melody Name	DCT		Fourier	
	MAD	MSD	MAD	MSD
test1.wav	0.176	0.306	0.174	0.306

test2.wav	0.167	0.343	0.166	0.276
test3.wav	0.172	0.381	0.173	0.304
test4.wav	0.173	0.347	0.168	0.275
test5.wav	0.076	0.234	0.072	0.110
test6.wav	0.170	0.376	0.163	0.256

IV. Conclusions

Several conclusions have been stimulated from the obtained test results.The retrieval process becomes faster when using short length audio query blocks, because the processing of feature extraction and matching stages take less computation time. For accurate retrieval, it is better to use a sufficient length of melody audio data to ensure good capturing for the discriminating features and consequently to make accurate retrieval decisions. In the proposed system, 30-35 second is a good query length to make better recognition. It is noticed that the execution time is significantly consumed in transformation (DCT and Fourier) stage; while the whole other processing stages takes less than (1%) of the total time. The use of optimizer to assess the suitable threshold value was a good idea to get a bounded number of energetic frequencies to be within certain range (e.g. [120,140] features). The limited number of frequencies is important to compute the hit count array that used to calculate the highest and most frequent (lived) frequencies; which are used as discriminating features. The average system performance achieved by our proposed system is 99.45%

References

[1] B. Shao, "*UserCentric Music Information Retrieval*", PhD dissertation, Florida International University, Miami, Florida, 2011.

[2] M. Müller, "*Information Retrieval for Music and Motion*", Germany, Springer-Verlag, ISBN 978-3-540-74047-6 Springer Berlin Heidelberg New York, 2007.

[3] C. d. Bandera, A. M. Barbancho, L. J. Tard´on, S. Simone and B. Isabel, "*Humming Method for Content-Based Music Information Retrieval*", 12th International Society for Music Information Retrieval Conference (ISMIR 2011).

[4] T. B. Mahieux, D. W. Ellis, "*Large-Scale Cover Song Recognition Using The 2D Fourier Transform Magnitude*", 13th International Conference on Music Information Retrieval (ISMIR 2012).

[5] J. Urbano, M. Schedl, X. Serra, "Evaluation in Music Information Retrieval ", Journal of Intelligent Information Systems, Vol 41, Issue 3, Pp 345-369, 2013.

[6] D. Salamon, "*Data Compression The Complete Reference*", Fourth Edition, Springer, ISBN-10: 1-84628-602-6, 2007.