International Science Index, Computer and Information Engineering Vol:1, No:4, 2007 waset.org/Publication/99

A New Spectral-based Approach to Query-by-Humming for MP3 Songs Database

Leon Fu, Xiangyang Xue

***Abstract*—**In this paper, we propose a new approach to query-by-humming, focusing on MP3 songs database. Since MP3 songs are much more difficult in melody representation than symbolic performance data, we adopt to extract feature descriptors from the vocal sounds part of the songs. Our approach is based on signal filtering, sub-band spectral processing, MDCT coefficients analysis and peak energy detection by ignorance of the background music as much as possible. Finally, we apply dual dynamic programming algorithm for feature similarity matching. Experiments will show us its online performance in precision and efficiency.

***Keywords*—**DP, MDCT, MP3, QBH.

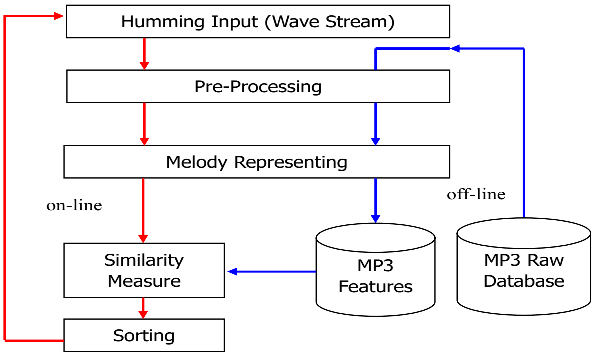
I. Introduction

WITH the rapid development of multimedia applications, we are enjoying more entertainment activities in our life especially the music songs. In order to meet the needs, more and more music information retrieval (MIR) systems have been developed. Some of them were based on text retrieval models by entering music names, genre, artists and etc while the others were based on content-based retrieval models by humming the melody or singing the lyrics. As we know, in many cases, we may not know the music profile information exactly; instead, we just remember some melody snippets so that we need to use the content-based models. However, there indeed exist many content-based retrieval methods but most of them were based on symbolic performance data such as MIDI streams, in which melody information is stored definitely and limited instruments are employed and also no vocal sounds are recorded. Since MP3 songs are much more widely used than symbolic ones in our common life, a new application of content-based retrieval system on MP3 songs becomes to be required. Therefore, in this paper, we introduce a new approach to make *QBH* (Query by Humming) work on MP3 songs.

MP3 is one of the most popular music formats, which is short for MPEG I/II Audio Layer 3. It is actually a sort of compressed and wave-formed polyphonic audio data which is encoded by psychoacoustic model, quantized signal processing and frame packing techniques so that all the instruments, vocal sounds and even noises have been recorded and integrated into it. Researchers have shown that it is so hard to separate these signals correctly and clearly in order to extract the rhythm information like symbolic performance data. Our approach is to extract the feature descriptors from frequency spectral information from the data streams. The difficulties will be shown in the further discussion.

Our retrieval system architecture is shown in Fig.1. It consists of two subsystems which are off-line module and on-line module. MP3 songs database collection and feature extraction are under off-line part while humming process and matching process are under on-line part. Therefore, all the MP3 raw data have been processed into descriptors and stored into feature database before humming is coming for query. And then our system converts the incoming humming queries into features as well and matches them with MP3 counterpart in database.

FIGURE 1
SYSTEM ARCHITECTURE



II. Previous Work

Most of the previous work on QBH are based on processing symbolic performance data and is focused on three components, which are melody extraction, melody representation and similarity measurement.

In melody extraction work such as [2][7], they use general method to extract pitches such as zero-crossing detecting, energy changing and median filters for humming process. The methods above are based on time domain analysis. As we know, symbolic data do not need melody extraction work so that it is only effective for discrete melody humming but not good for continuous melody humming and MP3 songs. Just in [5],

Manuscript received December 14, 2004. This work was supported in part by Natural Science Foundation of China under contracts 60402007 and 60373020, China 863 Plans under contract 2002AA103011-5, and Shanghai Municipal R&D Foundation under contracts 03DZ15019 and 03DZ14015, MoE R&D Foundation under contract 104075.

Leon Fu is with the Dept. of Computer Science and Engineering, Fudan University, China (e-mail: 022021221@fudan.edu.cn).

Xiangyang Xue is with the Dept. of Computer Science and Engineering, Fudan University, China. He is now the head of the Department of Computer Science (e-mail: xyxue@fudan.edu.cn)

MDCT (Modified Discrete Cosine Transform) is mentioned for MP3 but by manual segmentation. In our paper, we adopt to make use of MDCT coefficients analysis on frequency spectral to extract pitches adaptively from database and humming.

As to melody representation, [1] first presents to use only pitch contour <up/down/same> to represent melody. In [2], pitch interval and rhythm are both presented; in [3][4], four basic segment types and a triplet <time/pitch/beat> are considered to represent the melody. The entire above are relatively direct for symbolic data but not for real audio data. In this paper, we adopt to use quantized pitch change descriptor presented in our previous paper [6] but by new algorithm of peak energy detection from MP3 database.

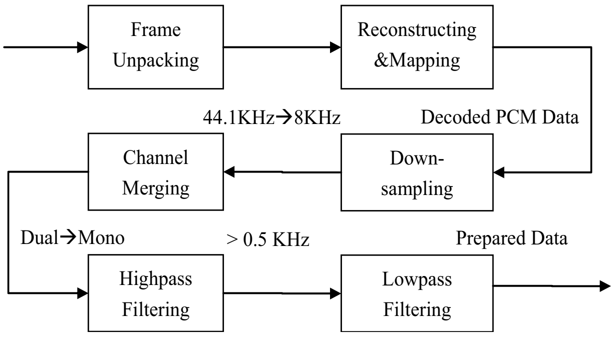
The matching algorithm for the similarity measure also needs to be considered. In [1], they use an approximate string matching algorithm described by Baesa-Yates and Perleberg. In [8], a new innovative distance metrics between query and songs is proposed based on symbolic data. [7] presents a "hierarchical matching" method which doesn't show its efficiency in response time. In a recent paper [9], they improve the existing DTW indexes technique by introducing envelope transforms which also did not show its efficiency in detail. In this paper, we adopt to use our proposed efficient dual-dynamic algorithm [6] for similarity measure.

III. MP3 Songs Melody Representation

A. Pre Process Work for MP3

MP3 songs not like symbolic performance data contain many complicated melody information and even noise disturbance, some of which are important for our melody representation while others we need to neglect. Thus we have to do some preprocess work for the MP3 songs to obtain what we really need. As we know, most of our recently used MP3 songs are 44.1 KHz sample rate and dual-channel data, but for our melody representation we do not need such high quality that it will make our further process time-consuming and inefficient. Actually even in very low sample rate we can also identify the melody of the songs. Therefore, we decode the MP3 into wave streams and down sample the raw MP3 songs to 8 KHz, single-channel. Although the quality is very poor for listening, it is enough for our following work.

FIGURE2
MP3 PRE PROCESS CHART

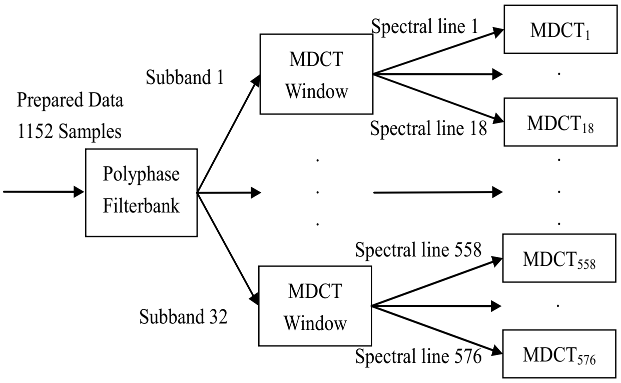


Since in MP3 songs, human vocal part always plays an important role in representing melody rather than its background music which is sometimes even a kind of beat rhythm. Furthermore, music researchers have shown that the most expressive instruments and euphonic vocal sounds are usually between 0.6 KHz and 2 KHz in their frequency band. In this case, we continue preprocessing the wave streams by lowpass filters and highpass filters. We first specify the highpass threshold as 0.5 KHz and then adjust the lowpass according to the audio data itself, it may be at 1.6 KHz or 1.8 KHz but not higher than 2.4 KHz. As to the humming, the preprocess work is in similar way. Fig.2 shows the flow chart.

B. MDCT Coefficients Extraction

MDCT is a *Modified Discrete Cosine Transform* used in MP3 encoder which has a perfect reconstruction performance. After preprocess work, we analyze the MDCT coefficients on frequency spectral subband from the prepared data. Fig.3 gives us the MDCT encoding structure in detail. From the figure, we can find out that the prepared data is handled by the polyphase filter bank which divides the frequency band into 32 sub bands in order to get better frequency resolution. Each subband window will do the transformation by 50% overlapping to get 18 spectral lines which we called MDCT coefficients. All the 576 coefficients represent the energy on different frequency base. Then, we apply butterfly algorithm on them to reduce the alias effect. Our features are extracted according to these processed values.

FIGURE 3
MDCT STRUCTURE



$$Eng_i = MDCT_i * MDCT_i$$

$$SP_k.EngAll = \sum_{i=1}^{576} Eng_i \qquad SP_k.EngMax = \max(Eng_i) \qquad (1)$$

$$SP_k.EngAvg = \sum_{i=k-L/2}^{k+L/2} SP_i.EngAll / L$$

where L is the window length , 18 in our experiments

C. Peak Energy Detection

Since the MDCT coefficients reflect the energy values, we can use them to produce the vocal melody for the songs. Here, we define *SP* (*sample point)* as a group of MDCT coefficients from 1152 samples. We also define *MP (melody point)* as the result from the following detection work. We first calculate the total energy and maximum energy by (1) and the main frequency value of each sample point is shown in (2). After that, we can detect the melody points we define as the peaks of energy change. The following procedure in Fig. 4 shows the peak energy detection algorithm.

$$SP_k.FreqMain = (MaxFreq_k - MinFreq_k)*i/576 + MinFreq_k$$
$$\text{where } Eng_i = SP_k.EngMax \text{ and } Eng_i > SP_k.EngAll*30\% \quad (2)$$

where $MaxFreq_k$ and $MinFreq_k$ are respectively the maximum and minimum frequency value in the specific MP3 window range. i is the spectral line number of MDCT coefficients where the main frequency is located.

FIGURE 4
PEAK ENERGY DETECTION ALGORITHM

```
Premise:
    1. AE (accumulative energy) = 0
    2. LSP (last sample point) = empty
    3: CMP (candidate melody point) = empty
    4: LES (last energy state) = UP
    5: SP_k (sample point) = the first fetched sample point
    6: CNT (count) = 1
Step 1: if (SP_k.EngAll < LSP.EngAll) goto 5
Step 2: if (LES == UP) goto 6
Step 3: if ( $AE > \sum_{i=k}^{k+CNT} SP_i.EngAvg*Threshold$ )
            CMP→melody point;
Step 4: LES = UP; AE=0; CNT=1; CMP = empty; goto 6
Step 5: LES = DOWN;
Step 6: AE += | SP.EngAll – LSP.EngAll |;
Step 7: if ( $SP_k.FreqMain * \sqrt{SP_k.EngMax} >$
            $CMP.FreqMain * \sqrt{CMP.EngMax}$ )
        CMP = SP_k
Step 8: LSP = SP_k; CNT ++;
        SP_k= next fetched sample point
```

This algorithm processing can be followed by the MDCT extraction procedure in pipelining since the time complexity is nearly in linear. In our experiments, we set the threshold in step 3 as 1.5, which means the accumulative energy needs to be higher than 1.5 times of the average accumulative energy. And in step 7, we check the peak energy due to the fact that higher frequency always keeps the dominant melody in the songs and higher energy always represents the vocal sound if exists. Therefore, we here consider the frequency value and its energy value both. Although we use moving window technique in the detection algorithm, but we do not divide the original data into individual segments due to keeping the relativity and integrity of our descriptors so that our following similarity measure can work efficiently.

D. *Descriptors Representation*

Since we can not ensure that exact same pitch is hummed and detected as the same, we eliminate the consecutive same melody points. In our descriptors we also ignore the duration time of the melody points since we also can not ensure that humming can keep better notes duration. If not, it sometimes even makes noises in similarity measure.

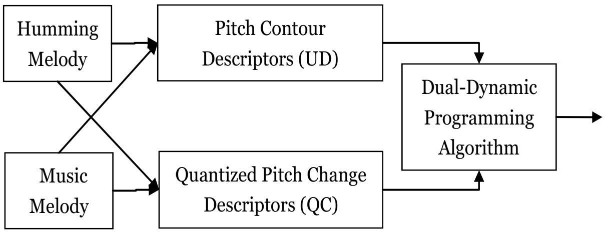
As peak sequence reflects the energy change on frequency and the frequency reflects the pitches or phoneme as well, we can apply our previous feature definitions [6] to represent our MP3 melody features. Pitch contour descriptor has been proved to be a simple but effective method for matching in practice and quantized pitch change descriptor also has well performance especially in error tolerance, which can ignore the tiny difference between two pitches in a small music scale which is shown in the following (3).

$$Q_k(Pitch_{k,i}) = \lfloor (Pitch_{k,i} - \min(Pitch_{k,0},...,Pitch_{k,l}))/\alpha \rfloor$$
$$QC_k(Pitch_{k,i}) = Q_k(Pitch_{k,i}) - Q_k(Pitch_{k,i-1}) \quad (3)$$

where k is the id of MP3 in database, i is the i-th melody point of k-th MP3, l is the length of the MP3 or humming, α is a quantizing step and in our system, it equals $1 + (\max_{i=0}^{l}(Pitch_{k,i}) - \min_{i=0}^{l}(Note_{k,i}))/8$. For humming query, it always has a changeable α step which always keeps it as same as that of each MP3 to be matched.

IV. Similarity Measure

FIGURE 5
SIMILARITY MEASURE PROCESS


Humming Melody
Pitch Contour Descriptors (UD)
Dual-Dynamic Programming Algorithm
Music Melody
Quantized Pitch Change Descriptors (QC)

Here, we use the proposed *dual-dynamic programming* algorithm [6] shown in Fig.5 with two DP (Dynamic Programming) units. As being proved, dynamic programming is an effective and efficient algorithm. Pitch contour descriptor still uses classic DP method as a unit while for the quantized pitch change descriptor we apply (4) for another DP unit which we adjust the error values by considering the distance between the two different quantized pitch change values instead of just adding one. There's a fact that when pitch changes more, human tends to make mistake more easily in humming so that we have to decrease the effect. Therefore in (4) the error value will be smaller if the pitch change T_j in database is bigger.

$$C'[0,j] = 0 \quad C'[i,0] = i$$
$$C'[i,j] = if\ ((|P_i - T_j| / (|T_j|+1)) < \beta)$$
$$then\ C'[i-1,j-1]\ else\ |P_i - T_j| / (|T_j|+1) + \quad (4)$$
$$\min(C'[i-1,j], C'[i,j-1], C'[i-1,j-1])$$

where P_i is the i th value of humming descriptor, T_j is the j -th value of database descriptor and β is a threshold.

As a result, for each humming query, we will get two error value matrixes C_{UD} and C'_{QC} to each MP3 in database. The last line of each matrix is what we need to calculate the error score for similarity using (5).

$$\begin{aligned} &Err_{k,i} = \gamma * C_{UD_k}[l1,i] + (1-\gamma) * C'_{QC_k}[l1,i] \\ &Sim(Hum, MP3_k) = \min(Err_{k,l1},\ldots,Err_{k,l2}) / l1 \end{aligned} \tag{5}$$

where $l1$ is the length of humming descriptor string, $l2$ is the length of k th MP3 descriptor string and γ is a weight, 0.3 in our experiments.

V. Experimental Results

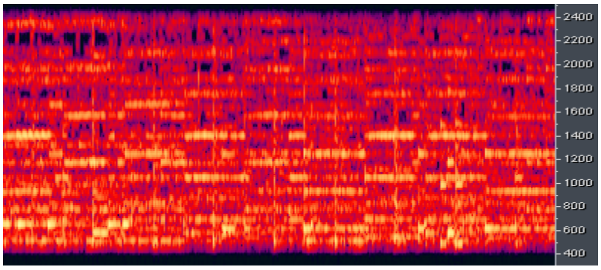
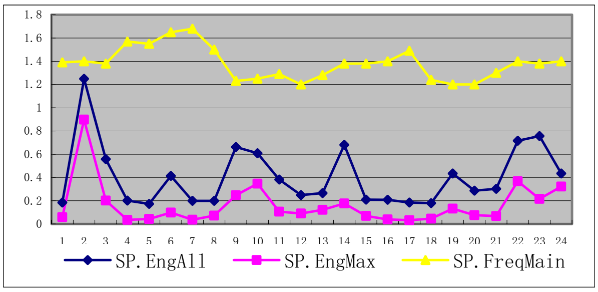


FIGURE 6
PART OF PRE PROCESS SPECTRAL VIEW

Our MP3 database was created by collecting 300 pop solo songs (150 of male vocal and 150 of female vocal) with total size of 1.05GB. Fig. 6 shows some spectral results after pre-processing with frequency between 400Hz and 2400Hz. The following MDCT extraction work is based on this spectral analysis. After applying peak energy detection algorithm on MDCT coefficients, we can extract the melody points from those main frequency curves. Fig. 7 shows us some extracted results, from which we can obtain the melody sequence as 1.4, 1.65, 1.25, 1.38, 1.2 and 1.4.

FIGURE 7
PART OF PEAK ENERGY DETECTION RESULTS

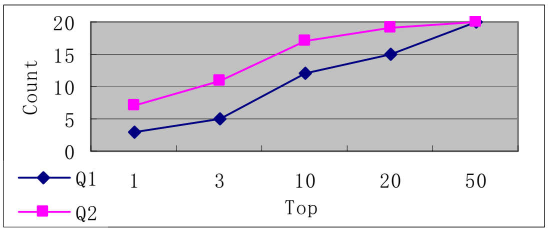


The humming pieces we prepared are for 20 different songs. All of these sources are hummed by those who have no music background. Most of the pieces are 10-15 seconds long that correspond to approximately 2 music phases of each song, from which we can extract 15-20 values for each descriptor.

In Fig. 8, we use one and two humming pieces for each query respectively (Q1 and Q2). Since our similarity measure is based on DP without segmentation, two humming pieces obviously can do some improvements in our experiments so that 7 queries can win the top 1 and 85% queries win the top 10. Practically, it is proved to be a feasible strategy in that it is acceptable for users to hum more than one pieces of a song for query. Although the pre-process and feature extraction time for MP3 songs database need take 10 seconds more time per each than symbolic data, the on-line response in spectral-based approach shows much more efficient than the temporal-based approach, which only takes 2.5 seconds per query from 300 songs.

However, according to the experiment results, there're still a few queries that do not show well performance and even ranks at 43rd. It is due to the fact that our method depends on the vocal part so that the background music should be as less effect as possible on the songs and most pop solo songs can keep it well.

FIGURE 8
QUERY TOP RANKS



VI. Conclusion

Query by humming for unstructured music database such as MP3 is a new application in content-based music retrieval area. But it still has some problems left to be solved such as non-vocal music, duet and chorus. Otherwise, the precision of experiment results also can be improved if we take times to combining some of temporal-based information. In this paper, we just present a new spectral-based approach to apply QBH efficiently on MP3 solo songs based on vocal part. We hope we can solve some other problems of this area in the further research work to make it work for all kinds of MP3 music.

References

[1] A. Ghias, et al, *"Query By Humming—Musical Information Retrieval in an Audio Database"*. Proc.s of ACM Multimedia95, pp231-236, 1995..

[2] R. J. McNab, et al, *"Towards the Digital Music Library: Tune Retrieval from Acoustic Input"*. Proc. of Digital Libraries, pp 11-18, 1996..

[3] A. L.P. Chen, M. Chang, J. Chen. *"Query by Music Segments: An Efficient Approach for Song Retrieval"*. In Proc. of IEEE International Conference on Multimedia and Expo., 2000.

[4] Y. Kim, W. Chai, R. Garcia, B. Vercoe, *"Analysis of a Contour -Based Representation for Melody,"* Proc. International Symposium on Music Information Retrieval, Oct. 2000.

[5] Chih-Chin Liu, Po-Jun Tsai, *"Content-based Retrieval of MP3 Music Objects"*, CIKM'01, 2001, Atlanta, USA

[6] Leon Fu, Xiang-yang Xue, *"A New Efficient Approach to Query by Humming"*, International Computer Music Conference 2004, ICMC, Miami, USA

[7] Lie Lu, Hong You, Hong-Jiang Zhang *A New Approach to query by humming in music retrieval*. Microsoft Research, China

[8] C. Francu and C. G. Nevill-Manning. *"Distance Metrics and Indexing Strategies for a Digital Library of Popular Music"*. In Proc. of IEEE International Conference on Multimedia and Expo. 2000.

[9] Yunyue Zhu, Dennis Shasha. *Warping Indexes with Envelope Transforms for Query by Humming*. SIGMOD 2003, June 9-12, 2003, San Diego, CA.