CONTENT-BASED RETRIEVAL OF POLYPHONIC MUSIC OBJECTS USING PITCH CONTOUR

Lihui Guo[1]*, Xin He*[2]*, Yaxin Zhang*[2]*, Yue Lu*[1]

[1]Department of Computer Science and Technology
East China Normal University, Shanghai 200062, China
[2]Motorola China Research Center, Shanghai 200041, China

ABSTRACT

This paper investigates the retrieval of content-based polyphonic music objects in Wav and MP3 format. The system allows user to find an intended song by humming or singing a section of it. In this paper we introduce the baseline system and describe the key components including the pitch extraction in humming/singing clip, the vocal/non-vocal music segmentation, the pitch tracking in polyphonic music, and the DTW based matching algorithm. We conducted evaluations on the system. The experimental results demonstrate the feasibility of retrieving polyphonic music objects by humming/singing.

Index Terms— Music information retrieval, pitch detection, query by humming/singing

1. INTRODUCTION

An emerging approach to music information retrieval is based on humming or singing. Query by humming/singing (QBHS) system allows the users to find a song by humming the tune or singing the lyric they know. The system takes the acoustic input and queries the songs that contain the humming/singing tune from the music database. Finally, the system outputs a ranked list of the candidate songs. Since no textual input is needed, query by humming/singing is a natural manner for music information retrieval system. This kind of systems would help people access to the interested music conveniently. Traditional retrieval ways based on text can be totally avoided.

In contrast to the retrieval of musical scores based MIDI music which is easy to acquire the melody information by the selection of the symbolic tracks, the retrieval of acoustic signal based polyphonic music, such as Wav and MP3, requires to extract the main melody information from the instrument accompanied singing voice. Since polyphonic audio melody extraction is well-known as a tough task, previous studies [1] on query by humming/singing had almost focused on monophonic MIDI music. Due to the popularity of the polyphonic music, it is far more practical to retrieve Wav/MP3 objects than MIDI. Liu [2] proposed a system that used polyphase filters to compute the MP3 features for indexing the MP3 objects. In his system, the MP3 songs should be manually segmented into phrases first. The manual segmentation is also required in Yu's system [3]. Another system proposed by Lie et al. [4] employed the MDCT spectral coefficients to represent the tonic characteristic of a short-term sound. The work by Doraisamy et al. [5] presented a polyphonic music retrieval system. However, their system was still based on MIDI music database. To our knowledge, no comprehensive query by humming/singing approach to retrieve Wav/MP3 objects has been reported in the literature.

This paper presents our investigation on retrieving the polyphonic music objects by humming/singing. The system lies on the pitch contours to find the similar song. It consists two phases. The offline phase is concerned with the construction of pitch feature database for the polyphonic songs. In the online phase, the system extracts the pitch sequence of the humming/singing query and computes the similarity distance between the input query and the songs in music database. The details of each stage are given below.

2. PITCH DETECTION OF HUMMING/SINGING QUERY

The goal of this stage is to capture the pitch sequence of the humming/singing query. Since pitch detector always obtains a false pitch value from the unvoiced frame, the input signal is passed through the voice activity detection (VAD) to detect the voiced frames. In this paper, we proposed a spectral entropy order statistics filtering (OSF) based VAD. First, an FFT is applied to compute the spectral magnitude X_i. Then we use the following formula proposed by Jia [6] to calculate the probability density function.

$$p_i = (X_i + C) / \sum_{j=0}^{N-1} (X_j + C) \;\; (0 \leq i \leq N-1) \qquad (1)$$

where p_i is the spectral probability of the i^{th} frequency component, and $N = 512$ is the frame length. C is a positive constant. The humming/singing signal may be corrupted by the user-generated artifacts such as heavy breathing, mouth noise, etc. These effects can be alleviated by the introduction of C [6]. The negative spectral entropy of the l^{th} frame is $H_l = \sum_{i=0}^{N-1} p_i \log p_i$. In order to improve the robustness of VAD, the proposed algorithm formulates the voice/unvoice decision rule by using an OSF [7] on the long-term spectral entropy information. Here, the implementation of OSF is based on $2m+1$ spectral entropies $\{H_{l-m}, ..., H_l, ..., H_{l+m}\}$ around the frame to be analyzed. For the initialization of the algorithm, the first m frames of the input signal are assumed to be unvoiced. The spectral entropy order statistics are obtained by sorting $\{H_{l-m}, ..., H_l, ..., H_{l+m}\}$ in ascending order. $H_{(r,l)}$ is defined as the r^{th} largest number of this sequence. The spectral entropy OSF estimator is measured by:

$$E_l = (1-\theta)H_{(r,l)} + \theta H_{(r+1,l)} \quad (2)$$

where $r = \lfloor \alpha L \rfloor (L = 2m+1, 0 < \alpha < 1)$, $\theta = \alpha L - r$. If E_l is greater than a threshold η, the l^{th} frame is classified as voiced frame, otherwise it is classified as unvoiced frame. η is formulated by $\eta = \beta H_{mid}$ $(0 < \beta < 1)$. H_{mid} is the median value of the set $\{H_0, ..., H_{m-1}\}$.

If a frame is classified as voiced, the modified autocorrelation method [8] is used to detect the pitch. Otherwise, it is dropped. In order to eliminate the undesirable pitch points, the detected pitch will be discarded if it is not in a plausible pitch range (80-500Hz). Because strong second harmonics can produce pitch-doubling effect, the pitch sequence may not be smooth enough, we apply a median filter of order 5 to the pitch sequence, which will be used in the matching procedure.

3. PITCH TRACKING OF POLYPHONIC MUSIC

In this stage, we aim to extract the pitch track of the singing voice from polyphonic song. There are two subtasks in this stage, the singing voice detection and the predominant pitch detection. If the polyphonic song is in MP3 format, it's first decoded to mono-channel Wav and downsampled to 16 kHz.

3.1. Singing voice detection of polyphonic music

For a given song, singing voice detection is to partition the polyphonic song into different portions, and classify each portion as being of either the pure instrumental type (non-vocal portion) or as a mixture of singing voice with/without background instrumental accompaniments (vocal portion). We locate the vocal/non-vocal segments by using an onset detection function that calculates the distance between the spectral magnitudes of target and observed frames [9]. First, an STFT is applied to each frame of the input signal. To better quantify percussive and tonal onsets, an Euclidean distance between target and observed STFT is calculated in complex domain:

$$D[t] = \frac{1}{N} \sum_{k=0}^{N-1} \parallel \hat{X}_k[t] - X_k[t] \parallel^2 \quad (3)$$

where t is the frame index, N is the length of STFT. $X_k[t]$ is the observed spectral of the k^{th} bin at frame t. $\hat{X}_k[t] = |X_k[t]| e^{j\hat{\phi}_k[t]}$ is the target STFT of the same frame and the same bin. $\hat{\phi}_k[t]$ is the phase deviation function, derived as:

$$\hat{\phi}_k[t] = \text{princarg}(\frac{\partial^2 \phi_k[t]}{\partial t^2}) \quad (4)$$

where princarg maps the phase to the $[-\pi, \pi]$ range, $\phi_k[t]$ is the corresponding phase of $X_k[t]$. In order to select the onsets independently of the current context, a dynamic threshold, $\vartheta[t]$ is calculated by:

$$\vartheta[t] = \lambda_0 D_{mid}[t] + \lambda_1 D_{avg}[t] \quad (5)$$

where $D_{mid}[t]$ is the median value of $\{D[t-b], ..., D[t+a]\}$ and $D_{avg}[t]$ is the average value of $\{D[t-b], ..., D[t+a]\}$. a and b define the window of the detection points considered, Generally, one frame in advance and five frames in the past [9]. λ_0 is the scaling factor and λ_1 is a positive proportion factor. Then, the detection function is formulated as $\varphi[t] = D[t] - \vartheta[t]$. A frame t will be selected as an instance of the musical onset if $\varphi[t]$ is the *maximal peak* in $\{\varphi[t-q], ..., \varphi[t], ..., \varphi[t+q]\}$. q is the minimal interval between two neighboring onsets. The position of selected onset is the portion boundary.

After the polyphonic song is partitioned into portions, we pool the information of each portion to make the vocal/non-vocal decision. We evaluate the probability that a portion is vocal or non-vocal by using vocal GMM, G_v, and non-vocal GMM, G_{nv}, respectively. Assumed that $\{F_1, ..., F_i, ..., F_M\}$ is the feature vectors of a portion with M frames, the *vocal probability* is represented by $\sum_{i=1}^{M} \log p(F_i|G_v)$, and *non-vocal probability* is represented by $\sum_{i=1}^{M} \log p(F_i|G_{nv})$. A portion is classified as vocal if $\sum_{i=1}^{M} \log p(F_i|G_v)$ is greater than $\sum_{i=1}^{M} \log p(F_i|G_{nv})$, vice versa. We select LPC-derived mel-cepstral coefficients (LPMCCs) [10] as the classification feature. A 15-dimensional LPMCC feature vector F_i is calculated from each frame.

3.2. Predominant vocal pitch detection

After the singing voice detection, each vocal portion undergoes a predominant pitch detector to detect the pitch track of singing voice. In our system, the predominant pitch detection is extended from the one used in Li's singing voice separation system [11]. The predominant vocal pitch detection algorithm detects multiple, simultaneous pitch tracks from vocal portion. Compared with the original algorithm, we improve the estimated distribution of relative pitch for singing voice.

First, the vocal portion is passed through a 128-channel gammatone filterbank. Channels with center frequencies lower than 800 Hz are designated as low-frequency channels. Others are designated as high-frequency channels. For high frequency channels, we extract the envelope of filter output. A

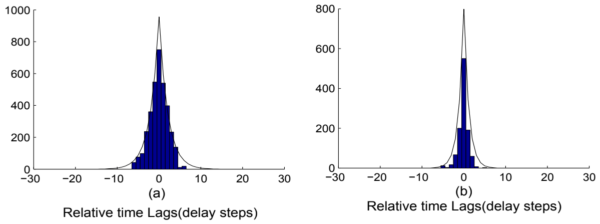


Fig. 1. Histogram and estimated distribution of the relative time lags of one pitch of a single channel. (a)singing voice; (b) speech.

normalized correlogram [12] is calculated for each channel with a frame length of 16 ms and the frame shift of 10 ms to obtain the periodicity information. Then, a low-frequency channel is selected if the maximum value of its normalized correlogram in plausible pitch range (80-500Hz) is greater than a threshold $\theta_1 = 0.935$. The time lags of peaks in selected channel are included in the set of peaks ϕ. In the high frequency range, all high-frequency channels are retained and only the first peak at a non-zero lag is selected and added to ϕ. The observation probability of 1-pitch hypothesis d for channel c is derived as:

$$p(\phi_c|d) = \begin{cases} q_1(c)U(0;\eta_c) \text{ if channel } c \text{ is not selected} \\ (1-q)L(\delta;\lambda_c) + qU(\delta;\eta_c) \quad \text{else} \end{cases} \tag{6}$$

where ϕ_c is the set of selected peaks in channel c. δ is the relative time lag between d and the closest selected peaks l. $L(\delta;\lambda_c)$ is the distribution of δ, which is described by a Laplacian distribution [11]. λ_c is the Laplacian distribution parameter. Fig.1.(a) shows the estimated distribution of δ from singing voice. Fig.1.(b) is estimated from speech, which is used in [11]. As can be seen, the distribution of δ from singing voice drifts down more slowly and spreads more widely than speech. There are a number of significant differences between speaking and singing in terms of both production and perception. An important difference is "formant tuning", the modification of the vocal tract to change the location of formant when singing [13]. Hence, the closest selected peak l may belong to the singing formant in some cases. $U(\delta;\eta_c)$ is a uniform distribution used to model the background music. q is the partition factor. $q_1(c)$ is the parameter q for channel c estimated from one-pitch frames. η_c is possible range of the distance of the two lags. Then the observation probability of the 1-pitch hypothesis across all channels is calculated by:

$$p(\phi|d) = k \sqrt[b]{\prod_{c=1}^{C} p(\phi_c|d)} \tag{7}$$

where ϕ is the set of all selected peaks. C is the number of channels. k is the normalization factor. b is the smoothing factor, which is used to compensate for statistical dependency among channels [12].

The observation probability of 2-pitch hypothesis, d_1 and d_2, in channel c is formulated by:

$$p(\phi_c|d_1,d_2) = \begin{cases} q_2(c)U(0;\eta_c) & \text{if channel } c \text{ is not selected} \\ p(\phi_c|d_1) & \text{if channel } c \text{ belongs to } d_1 \\ \max\{p(\phi_c|d_1), p(\phi_c|d_2)\} \text{ else} \end{cases} \tag{8}$$

$q_2(c)$ is the partition factor for channel c under 2-pitch hypothesis. Channel c belongs to the source d_1 if the distance between d_1 and the closest peak in channel c is less than $5\lambda_c$. All these parameters are obtained from singing voice using the maximum likelihood method in a manner similar to [11]. The frame observation probability of the 2-pitch hypothesis across all channels has the same form as Eq. (7).

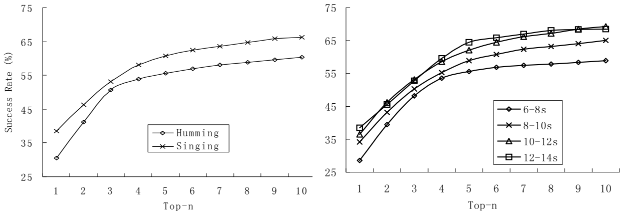
Finally, we employ an HMM to model the pitch generation process. The pitch state space is a union of Ω_0, Ω_1, and Ω_2, each of which represents the collection of hypotheses with zero, one, and two pitches. In each frame, the hidden node indicates the pitch state space, and the observation node represents the set of observed peaks ϕ. The observation probabilities have been formulated. The pitch transition between consecutive frames, i.e., between different pitch states is described by pitch dynamics. The state transition probability is once again described by a Laplacian distribution [12]. All these probability can be determined by training. Then the Viterbi algorithm is used to decode the optimal sequence of the pitch states. The first detected pitch track is considered as the pitch of singing voice.

4. MATCHING PROCEDURE

Because note segmentation is not performed in our system, our matching engine concentrates on aligning the query and target directly at the frame level. In this paper, we use the DTW algorithm to compute the similarity score between the pitch sequences of humming/singing query and polyphonic song. The DTW matching procedure is similar to the one used in [1]. Since the DTW is a classic and widely used method, we would not present its details here. Finally, the system returns a ranked list of the similar songs according to their DTW similarity scores.

5. EVALUATION

In this section, we present the evaluation of the proposed system. The music database used in this study consists of 35 polyphonic songs, including jazz, rock, country, R&B, etc. As for the humming/singing database, 400 humming/singing clips from 20 gender balanced singers without music background were recorded in an office environment. The singers can sing the lyric or hum the melody begin from and end at anywhere of the song.



(a) (b)

Fig. 2. The retrieval success rate on matching ranks.

The performance of the system is assessed in terms of top-N success rate, which is calculated as the percentage of the humming/singing queries that their correct songs can be found among top-N ranking. First, we categorize the clips into two subsets: the humming and the singing, to represent different query manners. We investigate the effects of the query manner on the system performance. Fig.2.(a) shows the top-10 success rate on the two subsets. It obviously shows that the success rate on singing subset is higher than humming subset. For most singers, singing the lyric is a natural way for querying the intended songs, while humming "LaLaLa" or "DaDaDa" may make them difficult to keep the exact rhythm. However, since some singers may not remember the lyric and only knows the melody, humming would be the only way to query a song.

We have also investigated the effects of humming/singing duration on the system performance. In the humming/singing database, the duration of humming/singing clips ranges from 6s to 16s. We divide the humming/singing database into four subsets according to their durations. Fig.2.(b) shows the evaluation results in different duration subsets. As can be seen, the top-1 success rate is increased with the increase of humming/singing duration. In some cases, the match engine can find the similar songs more exactly with a longer duration. In music database, some songs may have a part of similar melody, the discrimination ability would be improved with more melody information input. However, the overall relative improvements become less obviously when the duration is greater than 10s. With the duration of 12-14s, the correct query rate is 37% for the best one candidate and 50% for the best 3.

6. CONCLUSION AND FUTURE WORK

This paper presents a general framework for content-based polyphonic music retrieval system, which allows user to search the intended songs by humming/singing. We have developed a baseline system and performed experimental evaluation. The experimental results demonstrate the feasibility of retrieving the polyphonic music objects by humming/singing. To further raise the query success rate, the future work will be mainly focused on the two bottlenecks, the accuracy improvement of the vocal music segmentation and the polyphonic music pitch extraction.

7. REFERENCES

[1] J.S.R. Jang, N.J. Lee, and C.L. Hsu,"Simple But Effective Methods for QBSH at MIREX 2006", in *Proceeding of ISMIR*, October 2006.

[2] C. Liu and P. Tsai, "Content-based retrieval of MP3 music objects", in *Proceeding of ICIKM*, 2001, pp.506-511.

[3] H. Yu, W. Tsai, and H. Wang, "A query-by-sing technique for retrieving polyphonic objects of popular music", in *Proceeding of AIRS*, October 2005, pp. 439-453.

[4] W. Lie and C. Su, "Content-based retrieval of MP3 songs based on query by singing" in *Proceeding of ICASSP*, May 2004, pp.V.929-932.

[5] S. Doraisamy and S. Ruger, "Robust polyphonic music retrieval with N-grams", *Journal of Intelligent Information Systems*, vol. 21, pp. 53-70, 2003.

[6] C. Jia and B. Xu, An Improved entropy-based endpoint detection Algorithm", in *Proceeding of ISCSLP*, December 2002, pp. 285-288.

[7] J. Ramirez, J.C. Segura, et al., "An Effective subband OSF-based VAD with noise reduction for robust speech recognition", *IEEE Trans. on Speech and Audio Processing*, vol. 13, pp.1119-1129, 2005.

[8] L. Rabiner, et al.,"A Comparative Study of Several Pitch Detection Algorithms", *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-24, pp. 399-417, 1976.

[9] P. Brossier, J.P. Bello, and M.D. Plumbley, "Real-time temporal segmentation of note objects in music signals", in *Proceeding of ICMC*, November 2004.

[10] H. Fujihara, T. Kitahara, M. Goto, K. Komatani, T. Ogata, and H.G. Okun, "F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and Viterbi search", in *Proceeding of ICASSP*, May 2006, pp.V.253-256.

[11] Y. Li and D.L. Wang, "Separation of singing voice from music accompaniment for monaural recordings", *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 15, pp. 1475-1487, 2007.

[12] M. Wu, D.L. Wang, and G.J. Brown, "A multipitch tracking algorithm for noisy speech", *IEEE Trans. on Speech Audio Process*, vol. 11, pp. 229-241, 2003.

[13] D. Hall, "How do they do it? The difference between singing and speaking in female altos", in *Proceeding of the 29th Penn Linguistics Colloquium*, February 2005.