Indexing and Matching of Polyphonic Songs for Query-by-Singing System

Tat-Wan Leung and Chong-Wah Ngo
Department of Computer Science
City University of Hong Kong
{csltw,cwngo}@cs.cityu.edu.hk

ABSTRACT

This paper investigates the issues in polyphonic popular song retrieval. The problems that we consider include singing voice extraction, melodic curve representation, and database indexing. Initially, polyphonic songs are decomposed into singing voices and instruments sounds in both time and frequency domains based on SVM and ICA. The extracted singing voices are represented as two melodic curves that model the statistical mean and neighborhood similarity of notes. To speed up the matching between songs and query, we further adopt proportional transportation distance to index the songs as vantage point trees. Encouraging results have been obtained through experiments.

Categories and Subject Descriptors: H.5.5 [Sound and Music Computing]: Methodologies and techniques. **General Terms:** Algorithms, Design, Experimentation, Human Factors. **Keywords:** ICA, Melodic Curve, Proportional Transportation Distance

1. INTRODUCTION

Content-based musical information retrieval (MIR) has recently attracted numerous research attention due to its potential commercial applications in Internet musical search and personalized musical retrieval system. Broadly, we can classify the research efforts in MIR to six different categories as shown in Table 1. In general, the modeling of polyphonic songs is more difficult than monophonic songs since the former involves multi-dimensional note representation and source separation. Similarly, the content analysis of symbolic representation (*e.g.,* MIDI) is usually more straightforward than waveform representation (*e.g.,* MP3). To date, most people believe that the problems in Type-I category are solvable although not very interesting in practice, while the issues from types IV to VI are hard-to-solve but have great commercial potential [8].

Table 1: Research in MIR

Type	Query	Database
I	M	M
II	P	P (MIDI)
III	P	P (Raw)
IV	M	P (MIDI)
V	M	P (Raw)
VI	M	P (Pop song, Raw)

M: monophonic, P: polyphonic

The issues we address in this paper belong to Type-VI where a user sings a small piece of song in few seconds from memory and the system retrieves similar popular songs. Type-VI can be viewed as a special case of Type-V since the predominant source in pop songs is the singing voice. An intuitive approach to this problem is to match the extracted singing voices from pop songs with user queries for retrieval. In this paper, we propose an end-to-end approach for pop song retrieval. The major components include:

- *Singing voice extraction.* A pop song is initially segmented by detecting the changes of acoustic features in time axis. The segments that contain singing voices (SV) are then located by a SVM classifier. The monaural ICA is further applied to extract the pure SV directly in frequency domain.

- *Melodic curve representation.* We propose two curves, namely NMC and NSC, to model the statistical mean and neighborhood similarity of notes for the extracted SV. In contrast to pitch contour representation that is popularly used in MIR, the proposed melodic curves are robust to noises induced by sound sources separation. The curves are represented as weighted point sets suitable for the time series matching.

- *Pop song indexing.* To speed up the matching of queries and songs in database, the proportional transportation distance (PTD) [7] is employed to index pop songs in vantage point trees. PTD, in contrast to dynamic time warping, is a pseudo-metric that can be applied directly to measure the distances of melodic curves for indexing.

Relatively few works [1, 6] were addressed for Type-VI category probably due to the technical challenge in blind source separation. In [6], a mid-level probabilistic note representation is proposed to model polyphonic songs, by avoiding the step of singing voice extraction. Because SV and IS are considered jointly for melodic extraction without source

MM'04, October 10-16, 2004, New York, New York, USA.

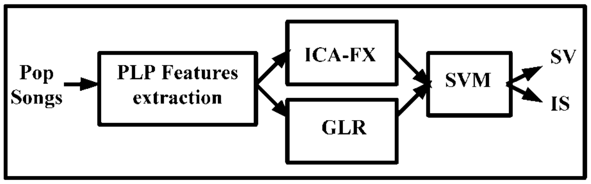


Figure 1: Temporal classification of SV and IS

separation, the note representation is vulnerable to background instrumental sounds and this could adversely affected the accuracy of matching. In [1], the stereo and monaural ICAs are applied for signal separation. Nevertheless, because the classification of IS and SV is not performed, the results of ICA on the segments purely containing IS are not analytical. In both [1, 6], the issues of song indexing for fast matching are not addressed. DTW and recurrent neural network are used by [1, 6] for query matching. PTD is originally used in [7] for the melodic similarity measure of musical scores. The effect of PTD on polyphonic raw audios is still unknown.

The remaining paper is organized as follows. Section 2 presents our approach in singing voice extraction. Section 3 describes the extraction and representation of melodic curves. Section 4 presents the issues in indexing and query matching. Section 5 shows the experimental results, and Section 6 concludes this paper.

2. POLYPHONIC SIGNAL SEPARATION

The fundamental problem of matching a monophonic query and a polyphonic song is the decomposition of polyphonic signals into independent sound sources. We tackle this problem in two separate steps: i) classification of singing voice (SV) and pure instrumental sound (IS) in time axis; ii) Extraction of pure SV from SV in frequency domain. Notice that the classified SV is not “pure” singing voices, but rather a mixture of singing voices and IS.

2.1 Classification of SV and IS

Figure 1 shows the flow of classification scheme. Initially, PLP (perceptual linear predictive coding) features (39-dimensional vector) are extracted from polyphonic songs. Generalized likelihood ratio (GLR) are then utilized to locate the boundaries of SV and IS based on the statistical changes of PLP features. At the meanwhile, ICA-FX is employed to transform and reduce the feature dimension of PLP. The segmented audio signals, as well as the extracted ICA-FX features from each segment, are ultimately input to the support vector machine (SVM) for pattern classification. We conduct experiments on 30 pop songs, and the results show that approximately 80% of SV segments are correctly located when the PLP feature vectors are reduced from 39 to 6 dimensional feature space. The details can be found in [4].

2.2 Monaural ICA

Denote $\mathbf{Y} = [y(1), y(2), \ldots, y(T)]$ as the non-pure SV, and let $\mathbf{X}_1$ and $\mathbf{X}_2$ as the time series of pure SV and IS respectively. Then, we can describe $\mathbf{Y}$ as a mixture of

$$\mathbf{Y} = \lambda_1 \mathbf{X}_1 + \lambda_2 \mathbf{X}_2 \tag{1}$$

For monaural songs, the goal is to recover the source $\mathbf{X}_1$ given only a single observation $\mathbf{Y}$. To solve this problem, we adopt the approach in [3] where a source is assumed as a linear superposition of basis functions with scalar multiples. The problem ultimately becomes the estimation of basis functions for $\mathbf{X}_1$ and $\mathbf{X}_2$ respectively from the pure SV and IS training data. A set of ICA filters based on the basis functions are then obtained for blind source separation (see figures 3 and 4).

3. MELODIC CURVE EXTRACTION

We propose two melodic curves, namely note mean curve (NMC) and neighborhood similarity curve (NSC), for the matching of queries and songs. Unlike pitch contours that are popularly used to represent monophonic songs, the proposed melodic curves are tailored specifically to the extracted SV by monaural ICA. The precise source separation is often impossible, as a result, the extracted SVs are not free of noise. Existing pitch tracking techniques can not be applied to robustly extract contours from such noisy signals.

To extract melody features, harmonic enhancement [6] is applied initially. The predominant sound in SV is extracted by selecting the outstanding peaks compared with the surroundings as follows

$$\mathcal{E}_t(k) = \sum_{i=-W}^{W} A(E_t(k) - E_t(k+i)) \qquad 0 \le k \le N \tag{2}$$

where $A(x) = x$ if $\forall x \ge 0$ and $A(x) = 0$ if $\forall x < 0$. In Eqn (2), N is the half window size of short-time Fourier transform, $E_t(k)$ is the energy at $\frac{samplefrequency}{2N-1}(k-1)$ of a SV spectrogram, and W is the size of support window.

One of the most important source in recognizing songs is the fundamental frequency. Because human vocal tract usually emphasizes the frequency band at the formant, it is not appropriate to use the maximum energy to detect fundamental frequency. Instead, we employ harmonic sum [6] that calculates the average energy of harmonics for each possible fundamental frequency p as follows

$$F_t(p) = \frac{1}{\lfloor N/p \rfloor} \sum_{n=1}^{\lfloor N/p \rfloor} \mathcal{E}_t(np) \tag{3}$$

To speed up the matching of $F_t(p)$, we quantize frequencies into bands based on the frequencies of musical notes represented in 12 notes per octave. The strength of fundamental frequency indexed by p in Eqn (3) is then transformed to a bin indexed by note m as

$$N_t(m) = \frac{\int_{L_m}^{U_m} F_t(p)dp}{|U_m - L_m|} \qquad 0 \le m \le M-1 \tag{4}$$

where L_m and U_m are respectively the lower and upper frequencies of note m. These parameters are calculated as follows: $U_m = N_m \times \beta$, $L_m = N_m \times \beta$, $N_{m+1} = N_m \times \alpha$, $\alpha = 10^{(\log 2)/12}$ and $\beta = 10^{(\log 2)/24}$. Notice that the average energy of harmonics is represented in logarithmic scale in order to mimic the frequency resolution of human auditory perception system.

Based on Eqn (4), the note mean curve (NMC) is computed as

$$\mu(t) = \frac{\sum_{m=0}^{M-1} N_t(m) \times m}{\sum_{m=0}^{M-1} N_t(m)} \tag{5}$$

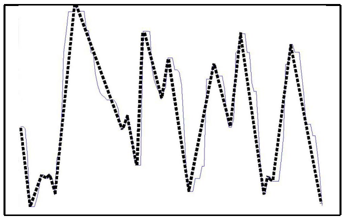


Figure 2: Results of curve trimming

which models the statistical mean of notes along the time axis, and at the meantime, smoothes certain amount of noises. Neighborhood similarity curve (NSC) is further computed to capture the cosine similarity of notes between two adjacent frames as follows

$$Sim(t) = \frac{\sum_{m=0}^{M-1} N_t(m) \times N_{t-1}(m)}{\sum_{m=0}^{M-1} N_t^2(m) \times \sum_{m=0}^{M-1} N_{t-1}^2(m)} \qquad (6)$$

Because the noise sources in two adjacent frames are normally similar, the angle between two note vectors will not be seriously affected. By encoding the cosine similarity between neighboring notes, NMC curve is robust to noises.

Since most users of query-by-singing are non professional singers, they cannot sing the same pitch as the original singers. To tolerate the possible noise in queries, we employ the curve trimming algorithm in [9] to capture the minimum and maximum points in NMC and NSC as the representation of curves. One example is given in Fig 2. The trimmed curved is represented by the dotted lines. After curve trimming, each slope in a curve is represented as $\mathcal{S} = \{\mathcal{P}, \mathcal{D}, \mathcal{L}\}$, where $\mathcal{P}$, $\mathcal{D}$ and $\mathcal{L}$ are respectively the starting point, direction and duration of a slope $\mathcal{S}$.

4. INDEXING AND RETRIEVAL

Matching a query and songs is usually a time consuming process. Typical singing queries from users last for about 10 sec, while the typical length of a popular song is around 3-5 min. The matching of a query and song by dynamic time warping (DTW), for instance, is a computationally intensive task. To speed up matching, an intuitive solution is to index each song in database prior to retrieval. Nevertheless, popular time series matching algorithms such as DTW are non-metric and thus can not be applied for song indexing. Instead, we employ proportional transportation distance (PTD) [7] which is a pseudo-metric and satisfies triangle inequality. Given a song, we employ breakpoint detection algorithm to partition the song into segments. A vantage point tree (VPT) [2] of the song is then constructed for indexing by measuring the PTD distances of segments.

4.1 Breakpoint Detection

The starting points of words are identified as the breakpoints to partition a song. The main reasons for this are: i) people do not start to sing at the middle of words; ii) the energy will usually increase sharply when people start to sing a word. Let the average power of a frame at t as

$$P(t) = \frac{\sum_{k=0}^{N/2} |E_t(k)|}{N/2+1} \qquad (7)$$

where N is the window size of short-time Fourier transform. The breakpoints are located at the local maximum points of the first derivative of $P(t)$. The spectrogram $E_t(k)$ is then partitioned into segments at the detected breakpoints. Each segment basically corresponds to one word. The NMC and NSC curves of each segment are then generated. The first 4 seconds of a segment is extracted for VPT indexing.

4.2 Proportional Transportation Distance

PTD measures the distance between two weighted point sets. In our case, the NMC (or NSC curve) of each segment are represented as a set of weighted points, and each point encodes the information of a slope $\mathcal{S}$. The weight of $\mathcal{S}$ is the duration $\mathcal{L}$ which will be flowed from one segment to the other. The distance between two points is defined in term of the starting point $\mathcal{P}$ and direction $\mathcal{D}$ of $\mathcal{S}$. Denote $\mathcal{S}_i = \{\mathcal{P}_i, \mathcal{D}_i, \mathcal{L}_i\}$ and $\mathcal{S}_j = \{\mathcal{P}_j, \mathcal{D}_j, \mathcal{L}_j\}$ as two points of the segments A and B respectively. The distance between $\mathcal{S}_i$ and $\mathcal{S}_j$ of a NMC curve is defined as

$$d_{NMC}(\mathcal{S}_i, \mathcal{S}_j) = \sqrt{\alpha(\mathcal{P}_i - \mathcal{P}_j) + (1-\alpha)(\mathcal{D}_i - \mathcal{D}_j)} \qquad (8)$$

where $0 \leq \alpha \leq 1$ is a weighting parameter to control the sensitivity between distances of starting points and directions. Denote $\mathcal{L}_{ij}$ as the weights flowed from $\mathcal{S}_i$ to $\mathcal{S}_j$ and let $\mathcal{F} = [\mathcal{L}_{ij}]$ as the set of all feasible flows. Then, by PTD, the distance between segments A and B of a NMC curve is

$$D_{NMC}(A,B) = \frac{\min_{F \in \mathcal{F}} \sum_{\mathcal{S}_i \in A} \sum_{\mathcal{S}_j \in B} \mathcal{L}_{ij} \times d_{NMC}(\mathcal{S}_i, \mathcal{S}_j)}{\sum_{\mathcal{S}_i \in A} \sum_{\mathcal{S}_j \in B} \mathcal{L}_{ij}}$$

The PTD distance of a NSC curve, D_{NSC}, is calculated in the same way.

4.3 Query Processing and Matching

Given a monophonic query, the entropy-based endpoint detection algorithm in [5] is applied to remove the silent segments. The query is then converted into NMC and NSC melodic curves. Usually most users cannot keep the same tempo as the original singers. To tolerate the temporal error, we scale the NMC and NSC curves with a scale factor $r = 1 + i/20$, where $-5 \leq i \leq 5$. The search of similar songs consists of two steps. In the first step, the first 4-second segment of a scaled curve is extracted to rapidly locate the initial position of potential candidates in VP-trees while filtering most of the false matches. In the second step, all the candidates are compared with the query again, but with the duration same as the query length scaled by different values of r. Notice that the length of a candidate usually covers multiple segments depending on query duration. Let $\mathcal{Q}$ as a query and $\mathcal{C}$ as a candidate, the similarity between $\mathcal{Q}$ and $\mathcal{C}$ is

$$Sim(\mathcal{Q}, \mathcal{C}) = \min_r \frac{L_d}{\beta \times D_{NMC}(\mathcal{Q}_r, \mathcal{C}_r) + (1-\beta) \times D_{NSC}(\mathcal{Q}_r, \mathcal{C}_r)}$$

where the subscript r represents the scale version of a curve and $0 \leq \beta \leq 1$ is a parameter to control the weights of NMC and NSC curves. $L_d = |\mathcal{Q}|/|\mathcal{C}|$ if $\mathcal{Q} \leq \mathcal{C}$, otherwise $L_d = |\mathcal{C}|/|\mathcal{Q}|$. The $|\mathcal{Q}|$ and $|\mathcal{C}|$ represent the number of slopes in $\mathcal{Q}$ and $\mathcal{C}$ respectively. The parameter $L_d \leq 1$ is used to control the degree of similarity when $|\mathcal{Q}| \neq |\mathcal{C}|$.

5. EXPERIMENT

We conduct experiments on a database of 46 popular songs. The average length of each song is about 4 minutes, and in total we have approximately 3 hours of songs. After applying the breakpoint detection algorithm, each song, on average, is partitioned into 450 segments. As a result, there are about 20,000 segments being indexed by the VP-trees.

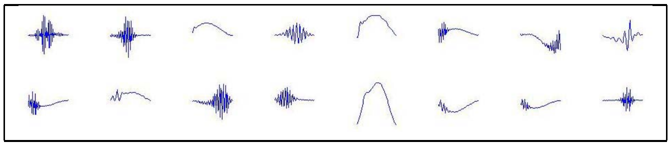


Figure 3: ICA basis functions of singing voices

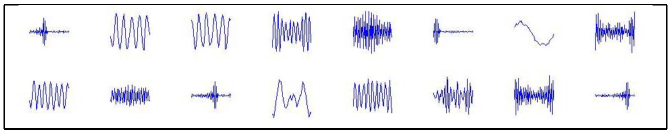


Figure 4: ICA basis functions of instrumentals

We collect 17 queries of different songs from 10 males and females for testing. Each person sings about 10 seconds. The main challenges in this experiment are: i) we use a tiny length query (10 sec) to match 20,000 segments in the database; ii) the matching needs to tolerate the errors caused by blind source separation. The retrieval results based on PTD are shown in Table 2. The empirical parameters we set are $W = 3$ (Eqn 2), $\alpha = \beta = 0.6$ (Eqn 8 and Eqn 9). In the table, top-n means the rate of queries that retrieves correct pop songs within top n rank. The experimental results indicate that the performance is reasonably good since around 64% of correct matches are ranked within top 5 position. Several correct answers are not included within the top 10 rank mainly because of the noise induced in signal separation. One correct answer is in very low rank because the beginning of corresponding segment in the pop song is mis-classified as instrument sound. As a result, the segment is not indexed in VP trees.

To contrast the performance of PTD and DTW, we conduct another retrieval experiment by DTW with the same set of queries. Because DTW is not metric, we do not utilize any index structure to speed up to the matching. The results are shown in Table 2. As indicated in the table, the results of PTD are better than DTW. In term of speed efficiency, the retrieval based of PTD is approximately 45 times faster than DTW. In Table 2, we also show the results of random selection as baseline comparison.

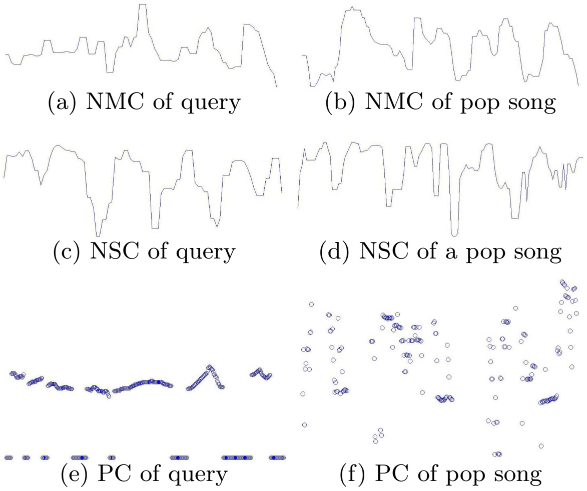
Table 2: Retrieval accuracy

Ranking	PTD	DTW	Random
Top-1	11.76%	5.88%	2.17%
Top-3	29.41%	23.52%	6.52%
Top-5	64.71%	52.94%	10.87%
Top-10	70.59%	64.71%	21.74%

Some intermediate results are shown in figures 3 to 5. In Fig 3 and Fig 4, the first few basis functions of singing voices and instrument sounds are given. In Fig 5, the NMC, NSC and pitch contours of a query and its corresponding segment in the pop song are shown. In contrast to the pitch contours generated by auto-correlation, the NMC and NSC curves of query and pop song are quite similar.

6. CONCLUSION

We have presented our proposed approaches in matching singing queries with polyphonic pop songs. Although the current database we use is still considered small, the problem we face is technically challenging. This is mainly because we use the complete songs (result in 20,000 segments), instead of small clips of songs for retrieval. Important findings from our works include: i) polyphonic signal separation is a hard yet important step in guarantee the success of our application; ii) the NMC and NSC curves are robust melodic representation for ICA extracted singing voices; iii) PTD is a good similarity measure for song indexing and query matching.



(a) NMC of query (b) NMC of pop song

(c) NSC of query (d) NSC of a pop song

(e) PC of query (f) PC of pop song

Figure 5: Comparison of NMC, NSC curves and PC (pitch contour) of a query and the corresponding segment in pop song.

Acknowledgments

The work described in this paper was partially supported by two SRG grants from City University of Hong Kong (Project No. 7001546 and 7001470).

7. REFERENCES

[1] Y. Feng, Y. Zhuang, and Y. Pan. Popular song retrieval based on singing matching. In *Int. Conf. on Music Information Retrieval*, 2002.

[2] A. W. Fu and et. al. Dynamic vp-tree indexing for n-nearest neighbor search given pair-wise distances. *Int. Journal on Very Large Data Bases*, 9(2):154–173, July 2000.

[3] G.-J. Jang and T.-W. Lee. A maximum likelihood approach to single-channel source separation. *Journal of Machine Learning Research*, 4:1365–1392, January 2003.

[4] T. W. Leung and et. al. Ica-fx features for classification of singing voice and instrumental sound. In *ICPR*, 2004.

[5] J. lin Shen, J. weih Hung, and L. shan Lee. Robust entropy-based endpoint detection for speech recognition in noisy environment. In *Int. Conf. on Spoken Language Processing*, 1998.

[6] J. Song and et. al. Mid-level music representation of polyphonic audio for query-by-humming system. In *Int. Sym. on Music Information Retrieval*, pages 133–139, 2002.

[7] R. Typke and et. al. Using transportation distances for measuring melodic similarity. In *Int. Sym. on Music Information Retrieval*, pages 107–114, 2003.

[8] C. Yang. Efficient acoustic index for music retrieval with various degrees of similarity. In *ACM Multimedia*, 2002.

[9] Y. Zhu, C. Xu, and M. Kankanhalli. Melody curve processing for music retrieval. In *ICME*, 2001.