MELODY CURVE PROCESSING FOR MUSIC RETRIEVAL

Yongwei Zhu, Changsheng Xu, Mohan Kankanhalli[†]

RWCP[*] Information-Base Functions KRDL Laboratory, Kent Ridge Digital Labs
21 Heng Mui Keng Terrace, Singapore 119613
[†]School of Computing, National University of Singapore
10 Kent Ridge Crescent Singapore 119260

ABSTRACT

There have been several query-by-humming techniques developed for music retrieval. The techniques either are error-prone due to the inaccuracy of the hummed query or force the users to hum according to a metronome. This paper presents a new slope-based query-by-humming technique, in which the retrieval is robust to the inaccuracy in query and the use of metronome is eliminated. We use *melody curve* to represent the melodies of the original songs and the hummed query. And curve features: *slope ranges and spans* and *note changes* are extracted from the melody curves. Music retrieval is done by matching the curve features of query with those of the original musical songs. Results have shown the features and the algorithms are robust to humming inaccuracy.

1. INTRODUCTION

With the proliferation of Internet and standardization of audio compression technology, there has been a great growth in number and size of music collections or databases. Conventional fashion of organization of music collection using singer's names, album's name, or any other text-based manner is becoming inadequate for effective and efficient usage of the music collection for average users. People sometimes prefer to access the music database by its musical content rather than textual keywords. Content-based music retrieval has thus become an active research area in recent years.

Since humming is the most natural way to formulate music queries for people who are not trained or educated with music theory. Therefore many researchers have proposed techniques for query-by-humming.

There are basically two approaches. In [1-5], pitch contour of the hummed query is detected and pitch changes are then converted into strings according to the direction and/or magnitude of the pitch change. Similarly, the melody contour of the MIDI music is also converted into strings, which are stored in the database. String matching algorithms are employed to do the similarity retrieval. In [6,7], user's humming is transcribed into MIDI melody using commercial software, and statistical features, such as note distribution, are used to match the query to the MIDI music files in the database.

The string matching approach requires precise detection of individual notes (onset and offset) out of the hummed query. However, it is not uncommon that people substitute a long note with several short notes with same pitch value while humming a tune. This may also occur in two renderings of a same melody. Furthermore, detection error will increase when tied notes are presented in the hummed melody. The string matching result would suffer drastically when such variation is not minor.

The second approach can cope with the above-mentioned issue. The query processing is done based on beats instead of notes. The statistical feature, such as tone distribution, is thus robust against erroneous query. This technique however requires the users to hum by following a metronome. Such requirement could be difficult for users sometimes. When a tune is hummed from memory, the user may not be able to keep in correct tempo. And different meters (e.g. duple, triple, quadruple meters) of the music can also contribute to the difficulty.

In this paper, we propose a new query-by-humming technique for musical song retrieval. The melodies of the musical songs are represented by melody curves and curve features are stored in the database. A user's hummed query is also transcribed to a melody curve, and curve features are extracted. The retrieval is done by searching for similar occurrences of the query melody curve in the database. Algorithms have been developed to extract robust curve features for the curve searching. This technique has overcome the limitation of forcing user to hum according to a metronome. User's inability of accurately following the original music's tempo is also tolerated.

This paper is organized as follows. Section 2 describes how to construct melody curves based on music scores and what features to be extracted from the melody curves. In section 3, we present algorithms, by which a hummed query is transcribed into melody curves and curve features are extracted. Section 4 presents a matching method for melody search, which is based on melody curve features. Section 5 concludes with a summary.

2. MELODY CURVE FROM MUSIC SCORES

2.1 Melody Curve

Melody in music is defined as a rhythmic succession of single tones organized as an aesthetic whole. Researchers have adopted melody contour to do music retrieval [1]. By melody contour, a melody is represented by a string with three alphabet symbols, such as "u" for up, "d" for down and "s" for same note, which correspond to the directions of the note changes. And music retrieval is conducted by doing string matching. Kosugi [6,7] utilized relative interval to increase the resolution of note changes.

We propose melody curve for melody representation and feature extraction. Melody curve is similar to melody contour in that, the horizontal dimension is time and vertical dimension is note/pitch value. The difference is that, there is no explicit note in melody curve and rest/silence note is substituted by

[*]Real World Computing Partnership

previous non-rest note. And the pitch values do not correspond to music keys. What is meaningful is the relative difference among the pitch values within one melody curve. Construction of melody curve from music score or MIDI music file is straightforward and is shown in next subsection. On the contrary the construction of melody curve for a hummed query is much difficult, and the algorithms for doing this will be discussed in section 3.

2.2 Melody Curve from Music Scores

Previous works have all focused on MIDI music for retrieval, mainly because of the wide availability of MIDI music files. However, all music records for real entertainment purpose are in wave format, such as CD audio or MP3 audio. Particularly, the singing voice in music songs can never be delivered by MIDI format. In our work, the melody information of the corpus is obtained from the music scores, although the technique would also works for MIDI music.

The construction of a melody curve from music score is straightforward. Each note in a music score corresponds to a piece of line in melody curve. Rests in music score are substituted by previous notes in the melody curve, so the melody curve can form a continuous line. The absolute pitch value in the melody curve is arbitrary. The lowest value is made sure to be above zero.

Figure 1 shows a part of the music score of a Chinese song “Qing Wang”. Its corresponding melody curve is illustrated in Figure 2(a).

Figure 1. A part of music score of Qing Wang.

In Figure 2(a), it can be seen that two adjacent notes with same key in the music score are connected together and form a horizontal straight line and rests are substituted by previous note in the melody curve. One semitone difference in music score corresponds to a difference of 10 in the melody curve shown in Figure 2(a).

2.3 Melody Curve Feature Extraction

The melody curve captures the main melodic information of the music, although identity of individual notes are discarded. We believe that the shape features of the melody curve can help to do robust melody search. We have identified a few important curve shape features: peaks and valleys, sharp value change.

A peak is a horizontal interval, at which the melody curve has a local maximum value. And a valley corresponds to a local minimum. They are shown in Figure 2(b).

We define the part of a melody curve from a peak to its next closest valley or from a valley to its next closes peak a *slope*. Each slope has a range value, which is the difference of the peak value and valley value. An up going slope has a positive range value and a down going slope has a negative range value. Each slope also has a span value, which is the time period from the start of the peak note to the start of the valley note, taking down going slope as an example.

Pitch value changes result from the note change in the music scores. Any two consecutive notes with different key value may correspond to a pitch value change in melody curve. The pitch value changes are shown in Figure 2(c).

Slope is the element that we used for curve matching. The value range and span of a slope and the pitch change values and note spans within a slope are thus important features that we will employ for doing curve matching.

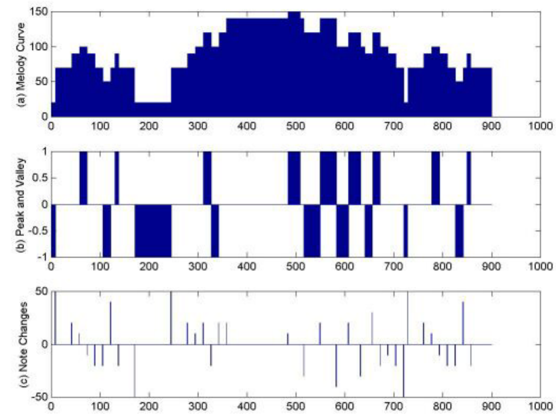

Figure 2. (a) Melody curve constructed from the music score; (b) the position of the peaks and valleys in the melody curve; (c) the pitch value changes.

3. MELODY CURVE PROCESSING FOR HUMMED QUERY

In our query processing, the hummed query undergoes a few steps of processing:

- Pitch tracking and melody curve construction;
- Melody curve trimming and peak valley detection;
- Pitch change detection.

3.1 Melody Curve Construction for Query

A classical pitch-tracking method using autocorrelation [8] is employed in our method.

Rest or silence in the query is detected by setting a threshold of the amplitude. The silence period in the query is replaced by the pitch value of the previous non-silence pitch. This curve is then logarithmically scaled down to make the vertical distance proportional to note distance used in melody curve. The vertical value is quantized into integer values with one octave corresponding a value range of 120, which means the vertical resolution is 1/10 of a semitone.

Figure 3(a) shows the pitch curve of a humming of the tune discussed previously.

3.2 Melody Curve Trimming

After the previous step, a rough melody curve is obtained. But the ubiquity of small variation in the curve has made the curve features such as peaks and valleys difficult to be extracted. Thus a melody curve-trimming algorithm has been developed to make the desired peaks and valleys obvious. The algorithm is stated as follows:

Each point in the melody curve can be treated as a mini-note with the span (length) of 1. If two consecutive points have a same value, then they can be treated together as a mini-note with the span of 2. And so on.

Set T_{span} to a value corresponding to the duration of an eighth note for moderate tempo.
For s = 1 to T_{span}
    For all mini-notes with span s and are also local maximum or minimum
        Combine this mini-note with its previous note or next note based on whichever is closer in pitch value, or if they all have the same pitch value, then all the three notes are combined. Then a new note with longer span is generated.

After identification of the peak and valley notes with span larger than Tspan, slopes are also identified. The pitch range of a slope is the difference between the peak and the valley, which are both ends of the slope.

Set the R_{slope_min} minimum pitch range of the slopes in the melody curve.
While $R_{slope\_min} < T_{slope}$
    Remove the slope with minimum pitch range by combining this slope with its previous and its next slopes. A new slope with larger pitch range is then generated. Find the slope with minimum range and reset R_{slope_min}.

T_{slope} is selected 10, which corresponds to the range of a semitone difference in pitch value.

After this algorithm finishes, the final peaks and valleys of the trimmed melody curve are identified.

3.3 Pitch Change Detection

Pitch change detection for the hummed melody curve is done based on the following algorithms:

For all detected slope
    Within a slope, calculate the pitch distance between every two adjacent mini-notes.
    Locate the minimum distance and combine those two mini-notes. The value of shorter notes is set to the value of the longer note. If the peak or valley note is involved, the peak or valley note conquers the other note.
    Repeat until the minimum distance is greater than threshold T_{slope}.
    The pitch changes are detected at any point in the melody curve that the pitch value is discontinuous.

After pitch change detection, a sequence of note span is also detected, which will also be used curve feature matching.

Result of query processing is shown in Figure 3 and 4. Figure 3 is the result for a humming by a male. Figure 3(b) shows the trimmed melody curve, and 3(c) shows the detected peaks and valleys of the melody curve. Detected pitch changes are shown in Figure 3(d).

From the result shown in Figure 3 and 4, it can be seen that the detection of the slopes in the melody curve is robust to humming errors. Compared with figure 2, there is even no misalignment of slopes. This shows slope is an appropriate element used for melody matching, which is discussed in next section.

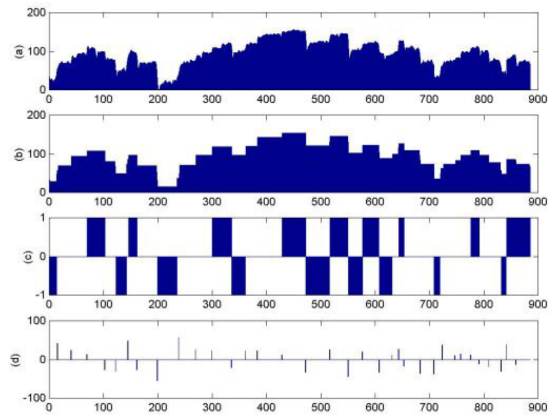

Figure 3. A hummed query by a male: (a) The pitch curve obtained by pitch tracking of a hummed query; (b) the melody curve after trimming; (c) the detected peaks and valleys; (d) detected pitch value changes.

Figure 4 shows the result for a humming by a female.

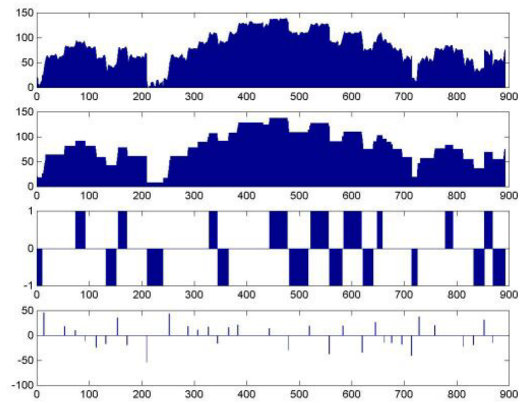

Figure 4. A hummed query by a female: (a) The pitch curve obtained by pitch tracking of a hummed query; (b) the melody curve after trimming; (c) the detected peaks and valleys; (d) detected pitch value changes.

4. SLOPE BASED MELODY CURVE MATCHING

In doing melody search, we propose a slope based melody curve matching method. A sequence of slopes, which are identified in the melody curve of a query, is searched in the database for similar occurrences. The features adopted are pitch value range of the slope S_R , span of the slope S_P , and the pitch value changes (N_{c1}, N_{c2}, …) and note spans (N_{p1}, N_{p2}, …) within the slope We denote the slope feature as (S_R , S_P : N_{p1}, N_{c1}, N_{p2}, N_{c2}, …) for each slope.

To match a sequence of n slopes to another slope sequence, 2 steps are taken: (1) slope sequence fitting; (2) melody contour matching.

In step 1, the slope sequences are matched by using only S_R and S_P. Those matched slope sequences are considered candidates and will be further matched in step 2. Two values are calculated in step 1: D_R and R_P using the following equations:

$$D_R = \left(\sum_{i=1}^{n} \left| S_R^H(i) - S_R^D(i) \right| \right) / n \qquad (1)$$

$$R_P = \left(\frac{\sum_{i=1}^{n} \left(S_P^H(i) \times S_P^D(i) \right)}{\left(\sum_{i=1}^{n} \left(S_P^H(i) \right)^2 \right) \times \left(\sum_{i=1}^{n} \left(S_P^D(i) \right)^2 \right)} \right)^{\frac{1}{2}} \qquad (2)$$

where n is the number of slope in the slope sequence for matching. $S_R^H(i)$ is the pitch range of the i^{th} slope in the humming query; and $S_R^D(i)$ is that for the candidate in database. $S_P^H(i)$ and $S_P^D(i)$ are the values of slope span.

D_R represents the average slope range difference for the two sequences, and R_P represents the correlation of the span of the two slope sequences. We set two thresholds: T_{DR} and T_{RP}. If $D_R < T_{DR}$ and $R_P > T_{RP}$, then a match is considered.

The matching is done for all possible candidates in the database, and all matched candidates will be furthered matched with the query in step 2. The computation of the step 1 is efficient, since only S_R and S_P are used.

In step 2, the contours of the two slope sequences are compared in more detail. Note value changes and note spans are employed to reconstruct the melody contour for further comparison. Many existing contour comparison methods can be adopted. We propose a contour-comparison-by-alignment method [9]. The main idea is to align the two contours horizontally by doing normalization and align vertically by computing the centroids of the two contours. The final similarity is the percentage of length of the query melody contour, where it has small distance (under a threshold) to the matching candidate after alignment.

In our experiments, we collected 80 music scores and 1,070 Karaoke file in MIDI format. The slope features are extracted and stored in the database. 5 users including male and female participate the experiments. They hummed the melodies of 3 difference songs through microphone. After pitch detection and slope feature extraction, the humming queries are searched in the database. The search results are ranked in a list ordered from high to low.

The results show that for 74% of the cases, the desired music is on the top of the rank list. And for 87% of the cases, the desired music is in the top-5 list. For more detailed experimental results, refer to [9].

The experiment results showed that the slope-based feature S_R and S_P are robust to humming errors or inaccuracy. And the slope sequence fitting method is essential for accurate and efficient matching.

5. SUMMARY

In this paper, we propose melody curve and curve processing technique for content-based music retrieval. A slope-based melody search algorithm is presented. Experiments show slope feature is robust to the errors in the hummed query.

Our slope-based melody search is superior to note-based [1-5] and beat-based [6,7] approach in that, it is robust to humming error and do not require the usage of metronome.

We believe the melody curve can facilitate in the development of an indexing structure for efficient retrieval of musical clips from a large database. We are currently working on this. In our future work, we will also incorporate tone distribution features into our melody contour matching, in which we can achieve a better retrieval result.

6. REFERENCES

[1] A. Ghias, J. Logan, and D. Chamberlin. "Query By Humming". *Proceedings of ACM Multimedia 95,* November 1995, pages 231-236.

[2] S. Blackburn and D. DeRoure. "A Tool for Content Based Navigation of Music". *Proceedings of ACM Multimedia 98,* 1998, pages 361-368.

[3] R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson and S.J. Cunningham. "Towards the digital music library: tune retrieval from acoustic input". *Proceedings of ACM Digital Libraries'96*, 1996, pages 11-18.

[4] P.Y. Rolland, G. Raskinis, and J.G. Ganascia. "Muisc Content-Based Retrieval: an Overview of the melodiscov Approach and System". *Proceedings of ACM Multimedia 99,* November 1999, pages 81-84.

[5] A. Uitdenbogerd and J. Zobel. "Melodic Matching Techniques for Large Music Database". *Proceedings of ACM Multimedia 99,* November 1999, pages 57-66.

[6] N. Kosugi, Y. Nishihara, S. Kon'ya, M. Yamanuro, and K. Kushima. "Music Retrieval by Humming". *Proceedings of PACRIM'99,* IEEE, August 1999, pages 404-407.

[7] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamanuro, and K. Kushima. "A Practical Query-By-Humming System for a Large Music Database". *Proceedings of ACM Multimedia 2000*, Los Angeles USA, 2000, pages 333-342.

[8] L.R. Rabiner, J.J. Dubnowski and R.W. Schafer. "Real-time digital hardware pitch detector". *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP 24(1):2-8, Feb 1976.

[9] Y.W. Zhu, M. Kankanhalli, and C.S. Xu. "Music Retrieval by Humming: A Slope-based Approach". Technical Report, Kent Ridge Digital Labs, 2001.