

SIMILARITY MATCHING OF CONTINUOUS MELODY CONTOURS FOR HUMMING QUERYING OF MELODY DATABASES

Yongwei Zhu, Mohan Kankanhalli*

Laboratories for Information Technologies, Singapore

* School of Computing, National University of Singapore, Singapore

ABSTRACT

Music query-by-humming is a challenging problem since the humming query inevitably contains much variation and inaccuracy. Many of the previous methods, which adopt note segmentation and string matching with dynamic programming, suffer drastically from the errors in the note segmentation which affects retrieval accuracy and efficiency. In this paper, we present a novel melody similarity matching technique which is based on continuous melody contour. The continuous melody contour has many advantages over previous string based melody representation. We introduce a contour alignment technique which addresses the robustness and efficiency issues. We also present a new melody similarity metric which is performed directly on continuous melody contours of the query data. This approach cleanly separates the alignment and similarity measurement in the retrieval process. We show how to robustly and efficiently align the query melody with the reference melodies and how to measure the similarity subsequently. Our melody alignment method can reduce the matching candidate to 1.7% with 90% correct alignment rate. The overall retrieval system achieved 80% correct retrieval in the top 10 rank list.

1. Introduction

Today tens of thousands of tunes are available in numerous music collections. The important and traditionally effective way for accessing the music is by the text labels attached to the music data, such as the name of singers or composers, title of the song or music album. But sometimes the text labels might not be characteristic of the piece or may not be remembered by users, and there is a need for accessing the music based on its intrinsic musical content, such as its melody, which is usually more characteristic as well as intuitive than the text labels.

Humming a tune is by far the most straightforward and natural way for normal users to make a melody query. Thus music query-by-humming has attracted much research interest recently. It is a challenging problem since the humming query inevitably contains tremendous variation and inaccuracy. And when the hummed tune corresponds to some arbitrary part in the middle of a melody and is rendered at an unknown speed, the problem becomes even tougher. This is because exhaustive

search of location and humming speeds is computationally prohibitive for a feasible music retrieval system. The efficiency of retrieval becomes a key issue when the database is very large.

Previous methods of music retrieval by humming are discrete music note-based approaches [1-7]. In the note-based approaches, a humming query is translated into a series of music notes. The note changes between adjacent notes are then represented by letters or symbols according to the directions and/or the quantity of the changes. The humming query is then represented by a string. In the database, the music notes of the MIDI files are also translated into strings in the same manner [1,2]. The retrieval is done by approximate string matching. Edit distance based on a dynamic programming approach is used for similarity measure. There are many limitations of this discrete note-based approach. It requires precise identification of each note's onset, offset, and note values. Any inaccuracies in the humming can lead to a large number of wrong notes detected and can result in poor retrieval accuracy. Furthermore, the performance of dynamic programming string matching cannot scale with the size of the database, since the search is done by brute force [1].

Jang in [7] presented a linear scaling method to cater for the humming speed variation issue. It uses 10 linear scaling variants from 0.75 to 1.25 for each reference melody candidate, which also means the size of the database is scaled up to 10 times. The efficiency issue is a major concern for this technique.

Kosugi et al. [8,9] have proposed a beat-based approach for query-by-humming. This method is robust against pitch variations in humming melodies. However, humming with a metronome is a rather restrictive condition for normal use. Many people usually are not very discriminating when it comes to their awareness of the beat of a melody. Different meters (e.g. duple, triple, quadruple meters) of the music can also contribute to the difficulties.

In this paper, we present an approach of doing melody retrieval based on a continuous melody contour representation. This representation is robust against variations in the hummed melody [10]. We invented a melody alignment method and a new melody similarity metric for melody contour matching. This technique separates the melody alignment and melody similarity measure into two stages, unlike the dynamic programming string matching methods which do it simultaneously. In stage 1, melody alignment method can efficiently locate the likely melody portion candidates and skips the unlikely candidates. This alignment method is robust against the errors or inaccuracy in the hummed query and is invariant to the humming speed of the query. After melody alignment, the melody similarity

measure is carried out in stage 2. The proposed similarity metric is for measuring the similarity of melodies using continuous contour representation. The robustness and efficiency of the alignment method, and the effectiveness of the similarity metric are demonstrated by the experimental results.

This paper is organized as follows: Section 2 presents the continuous melody contour for melody representation. Section 3 presents the proposed melody alignment method. Section 4, presents the melody similarity metric. Section 5 presents experimental results of the melody alignment and similarity metric. Section 6 concludes the paper with a summary and discussion.

2. Continuous Melody Contour and Melody Slopes

An appropriate melody representation is essential for effective melody retrieval. Previous symbolic approaches have been using note-based representation, where explicit music note is the basic unit. However, explicit music note identification from humming query is very error prone due to the errors or inaccuracy in humming. Studies [10, 12] have shown that a melody representation based on a continuous pitch contour without explicit note identity is adequate for capturing melodic information and possess substantial better precision for melody similarity measurement. The advantages of continuous pitch contour over string-based representation are: 1) variation of note identification in humming does not affect the representation; 2) better numerical precision of pitch values can be reserved in this representation, which can provide a much higher resolution of the melodic information than the string-based approach, which uses a semi-tone or larger pitch interval as the measurement unit.

Our query-by-humming music retrieval system has employed continuous melody contour representation. Both the melody for the MIDI files and the hummed queries are represented in continuous melody contours, and retrieval is based on similarity matching of the continuous melody contours.

To construct the continuous melody contour of a MIDI file, the monophonic music notes are firstly extracted from the MIDI file, and each note is converted to a horizontal line segments. The height of the line segments corresponds to the note value and the length of the line segment corresponds to note duration. In the continuous melody contour, two or more consecutive music notes with the same note value will correspond to one line segment and its length will correspond to the total duration of all the notes. When there is silence or rest in the melody, the duration of the silence is added to the duration of the previous non-silence note. As a result, the melody contour is a continuous sequence of line segments and it makes no distinction of each individual music note in the original melody.

In a continuous melody contour, the line segments corresponding to local maximum or local minimum, can be easily identified, and we call them peaks and valleys. We define the portion of a melody contour, starting from the beginning of a peak or a valley and finishing at the beginning of the next valley or peak, a *melody slope*. So a melody slope is a small portion of a melody contour, in which the note value changes only in direction, either increasing or decreasing. A melody slope has a

pitch range value, which is the note value difference between the corresponding peak and valley, and a *duration value*, which is the length of the melody slope. Figure 1 shows a continuous melody contour and its melody slopes.

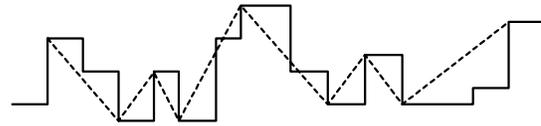


Figure 1: A continuous melody contour and melody slopes

When the humming query is input to the system, the first step of query processing is to find the pitch levels in the humming and construct a pitch contour. This is done by using a pitch tracking algorithm which we have developed [11]. The output of the pitch tracking algorithm, a pitch contour depicts how the pitch levels change in the humming query. The melody slope is difficult to identify, because the pitch contour contains many small turbulence. We use a pitch contour processing algorithm [10] to trim the pitch contour and transform the pitch contour to a sequence of horizontal line segments. After this processing, the peaks and valleys of the pitch contour become obvious, and melody slope of the humming melody can be identified. A special characteristic of our melody contour is that difference in pitch level within a melody contour for humming is measured at a higher resolution (1/10 of semitone), in order to retain the higher precision of the query data. If this is not done and the pitch level is measured in integral number of semitones, errors will be introduced which will severely degrade the quality of retrieval performance.

Experiments reported in [10] has shown that melody slope is much more robust against humming inaccuracy than note segmentation.

3 Melody Alignment Using Melody Slopes

Our melody retrieval system is based on the notion that the retrieved melody should have a melody contour very similar to that of the query melody. However, the query melody might be have rendered at faster or slower speed, and it may correspond to any part in the middle of the target melody. The corresponding melody contour should be aligned properly before a good similarity measure can be carried out. The alignment stands for temporally scaling the melody contour of a query melody and positioning it at certain locations of the reference melody contour. We assume the humming query is articulated at a consistent speed, so the variation caused by different humming speed can be compensated by proper temporal scaling. We propose a melody contour alignment method which utilizes melody slopes. This alignment method targets several objectives: 1) the alignment should be robust against inaccuracy or error in the humming, 2) the alignment also serves as the first step in matching which could reject, as many as possible, unlikely candidates without rejecting the desired candidate.

From our experiment of doing melody slope segmentation [10], the melody slope is not only robust against humming inaccuracy, but also illustrates a compact representation of the overall structure of the melody contour. We have found that a sequence of melody slopes can be used to achieve melody contour alignment with robustness and precision. The melody

alignment is essential for the similarity measure of melody contours.

The basic idea in the melody alignment is to locate the correspondents of a sequence of melody slopes of humming melody in the reference melody slopes. The pitch range values of melody slopes are used to determine whether two melody slope sequences potentially match. The duration values of the 2 melody slope sequences are used to compute a correlation value. The correlation value is high when the absolute speeds of the two melodies are different but are consistent. Details of the melody slope sequence search are presented in [11].

When there is a slope sequence match, the two slope sequences can be aligned by horizontally scaling the query melody slope sequence and make the total duration of the query melody slope sequence the same as the total duration of the reference melody slope sequence (figure 2).

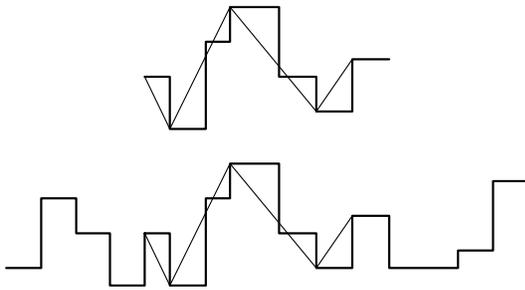


Figure 2: Horizontal alignment of 2 continuous melody contours

After horizontal alignment, vertical alignment is conducted. The basic idea is to compute the centroid of two melody contours, and use the centroid to vertically align the two contours and finally measure the difference of the two contours piece by piece. The process is presented as follows:

The melody contour of a query melody is denoted as $\{A(i), B(i)\}, i = 1 \cdots L_1$, and the melody contour of a reference melody is denoted as $\{C(j), D(j)\}, j = 1 \cdots L_2$. Where $A(i)$ is the pitch value of the i^{th} horizontal line segment in the query contour, and $B(i)$ is the length of the line segment. $C(j)$ and $D(j)$ are those values for the reference melody contour. L_1 and L_2 are the number of line segments in the melody contours respectively. We assuming the pitch value of the first line segment in the contour is 0.

Since these two contours have already been aligned to each other horizontally, what we need to do is to align them vertically. The centroid of the contours are computed respectively as

$$G_1 = \frac{\sum_{i=1}^{L_1} A(i)B(i)}{\sum_{i=1}^{L_1} B(i)} \quad (3)$$

and

$$G_2 = \frac{\sum_{j=1}^{L_2} C(j)D(j)}{\sum_{j=1}^{L_2} D(j)} \quad (4)$$

With the centroid of the contour, the two melody contours can be put together and distance between the line segments of the two contours can be calculated. This is illustrated in figure 3.

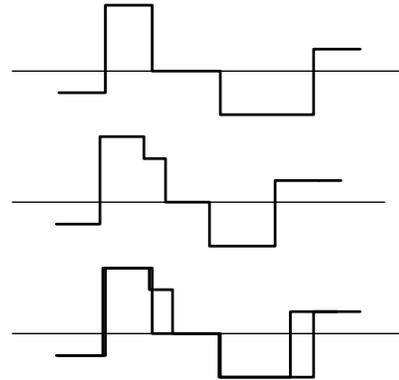


Figure 3: Vertical alignment of continuous melody contours

4 Melody Contour Similarity Metrics

Melody similarity metric is very important for effective melody retrieval. Especially when the size of database is very large, it is critical that the similarity metric has a good distinguishing capability and can retrieve the desired melody in the top 10 or top 20 rank lists. In our system, we directly compare the continuous melody contours of two melodies for the similarity measure, rather than converting the contour into symbolic string presentation and doing string matching. We propose a new melody contour similarity metric, which is performed after melody alignment.

After horizontal and vertical alignment, for each line segment in a query melody contour, a distance value can be computed. Any line segment in the reference melody contour that overlaps with this line segment, has a distance value. Among these values, the smallest distance value is kept for this line segment in query melody contour.

A similarity value for each melody slope in the query melody contour is computed as:

$$S_L(k) = \frac{P(k) - \sum_i |D_S(i)|}{P(k)} \quad (5)$$

where $P(k)$ is the pitch range of the k^{th} melody slope in the query melody contour, and $D_S(i)$ is the distance value of the i^{th} line segment in the melody contour. The summation of $|D_S(i)|$ correspond to the sum of the distance values of all the line segments within the melody slope. $S_L(k)$ is the similarity value for the k^{th} melody slope in the query melody contour. The maximum value of $S_L(k)$ is 1.

The final similarity value for the whole contour is computed as follows:

$$S_T = \sum_{k=1}^L S_L(k)W_L(k) \quad (6)$$

where L is the number of melody slopes in the query melody contour, and $W_L(k)$ is the weight of the k^{th} melody slope in the contour. $W_L(k)$ is computed as follows:

$$W_L(k) = \frac{N_L(k)}{N_T} \quad (7)$$

where $N_L(k)$ is the number of line segments in the k^{th} slope and N_T is the total number of line segment in the contour.

5 Experiments

We carried out extensive experiments to test our proposed methods for melody retrieval. We collected over 10,000 reference melodies, which are mainly MIDI files downloaded from the Internet. After melody feature extraction, 1,500,000 melody slopes are generated and stored in the database. We conducted two types of testing, one for melody slope alignment and the other for melody similarity measure.

5.1 Melody slope alignment

Since the melody slope alignment processing serves as locating likely candidate out of all possible candidates in the database. We did an experiment to find how many unlikely candidates are successfully rejected. The result shows that the more slopes are used, the less number of candidates are returned, which is obvious. When using 3-melody-slope-sequence, about 1.7% of all possible slope sequences are returned. The number returned by 4-melody-slope-sequence is even smaller, which is 0.5% of the population. Because a large number of slopes require the user to produce a very long query, which may not be feasible sometimes, we normally use 4-slopes-sequence in our retrieval system.

The other aspect of the melody alignment method is the robustness. This means the desired candidate should be contained in the returned candidates. We use 20 humming queries, each of which contains more than 10 slopes. The average rate of returned correct 4-slope-sequence candidate is 0.65. Since each query has multiple 4-slope-sequence, 1 single correct alignment is adequate for the proper similarity measure of the whole melody contour. The result shows 90% of the queries can find correct slope-sequence alignment.

5.2 Melody contour similarity metric

In the testing of the proposed melody contour similarity metric, we examine the retrieval accuracy of humming queries.

We use 4-slope-sequence for contour alignment. For each alignment, the total query melody contour is compared with the overlapped reference melody contours. 50 humming query melodies are used to test the overall retrieval result. We have achieved 80% correct retrieval in the top 10 rank list.

6 Conclusions

We have proposed a novel melody retrieval method by similarity matching of continuous melody contours, which consists of melody slope alignment and melody contour similarity measure. Unlike the previous method, such as the string matching approach, our approach separates the alignment and similarity measure into 2 distinct stages. The melody slope alignment stage aims to robustly locate the likely melody contour

candidate in the reference melody database efficiently and robustly. By rejecting large number of unlikely candidate, higher retrieval efficiency can be achieved. Experimental results demonstrated the performance of the method.

We proposed our melody contour similarity metric to achieve effectiveness in retrieval. The melody contour similarity measure is done after alignment. The alignment may be based on only a subset of the melody slopes in the considered melody contour. This metric quantifies the similarity of corresponding contours in their raw numeric representation. The metric exhibits high distinguishing capabilities. Satisfactory results of retrieval by humming justify the effectiveness of the metric.

Reference

- [1] A. Ghias, J. Logan, and D. Chamberlin. "Query By Humming". *Proceedings of ACM Multimedia 95*, November 1995, pages 231-236.
- [2] 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.
- [3] S. Blackburn and D. DeRoure. "A Tool for Content Based Navigation of Music". *Proceedings of ACM Multimedia 98*, 1998, pages 361-368.
- [4] P.Y. Rolland, G. Raskinis, and J.G. Ganascia. "Music 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] P. Y. Rolland, G. Raskinis, and J. G. Ganascia. "Music Content-Based Retrieval: an Overview of the Melodiscov Approach and System". In *Proc. ACM Multimedia 99*, November 1999, pages 81-84.
- [7] J.S. Jang, H.R. Lee, and M.Y. Kao. "Content-based Music Retrieval Using Linear Scaling and Branch-and-bound Tree Search", in *Proc. ICME 2001*, August 22-25 2001, pages 405-408.
- [8] 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.
- [9] 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.
- [10] Y. Zhu, C. Xu, M. Kankanhalli, "Melody Curve Processing for Music Retrieval", ICME 2001, Tokyo Japan.
- [11] Y. Zhu, M. Kankanhalli, C. Xu, "Pitch Tracking and Melody Slope Matching for Song Retrieval", PCM2001, Beijing China.
- [12] W.P. Birmingham, R.B. Dannenberg, etc, "MUSART: Music Retrieval Via Aural Queries", *Second International Symposium on Music Information Retrieval 2001*.