LOCAL ALIGNMENT FOR QUERY BY HUMMING

Qiang Wang, Zhiyuan Guo, Gang Liu, Chunguang Li, Jun Guo

Pattern Recognition and Intelligent System Laboratory
Beijing University of Posts and Telecommunications, Beijing, China
wangqiang086373@gmail.com

ABSTRACT

Query by humming (QBH) allows users to retrieve songs by humming a clip. In the previous work, the query has been regarded as a fragment of the music, so the task of QBH is considered to find a subsequence, which is most similar to the whole query, from the database. Taking into account humming errors, especially at the beginning or ending of the query, we assume that only part of the query is a subsequence of the music. Based on this assumption, we propose a local alignment framework which searches for the best match common subsequence between the query and database music. To verify the effectiveness of local alignment, two popular match algorithms, i.e. Linear Scaling and Dynamic Time Warping, are extended to identify the common subsequence. Experimental results on the 2010 MIREX-QBH corpus show that the new algorithms improve the retrieval accuracy significantly.

Index Terms— Query by humming, local alignment, music information retrieval, dynamic time warping, linear scaling.

1. INTRODUCTION

With the development of computer and network technologies, music data is growing rapidly, which has caused many new challenges in managing, exploring and retrieving music data. One of the challenges is how to provide a more convenient interface [1]. Query by humming (QBH), which can give users the wanted songs by only humming a few seconds, provides a natural interface for users in the situation that users only remember the melody, rather than the title or artist. However, it is a challenging problem for developers due to a variety of humming errors.

The interface of QBH was firstly put forward by Ghias et al. [2] in 1995. They searched songs by text retrieval techniques. Later, Jang et al. [3] introduced Linear Scaling (LS) algorithm which scaled the query to match the music. Frame based Dynamic Time Warping (DTW) [4], which can calculate the warping distance, was also introduced to QBH. Later, note based String Alignment (SA) was applied, which converted melodies into notes and then used dynamic programming techniques to align two note strings [5]. In 2012 Qiang et al. designed a multilayer filter for query by humming based on tempo variation [6]. Guo et al. took advantage of both melody and lyric to match songs [7]. To improve the retrieval efficiency, Locality Sensitive Hashing (LSH) [8] [9] [10] was introduced to QBH [11]. LSH is an index based fast neighbor search algorithm. Inspired by this indexing strategy, several variants of LSH were presented later. For example, Yu et al. suggested a new multi-stage LSH scheme [12] and Guo et al. proposed pitch and note based LSH indexes [13] [14] to search for candidate songs.

All the previous studies have been based on an assumption that the whole query is part of a song. However, due to humming errors, just part of the query is hummed correctly. In such a case, only part of the query is a subsequence of the song. If we search for the pieces that best match the whole hummed clip, humming errors can badly reduce the similarity score of the target song. To solve this issue, we propose a local alignment framework, which aims to identify the common subsequence with the largest similarity between the query and database songs. In the framework, part of the query could be discarded if it contains serious humming errors, especially at the beginning or ending of the query. Based on such a framework, we implemented two algorithms: Local Alignment Linear Scaling (LALS) and Local Alignment Dynamic Time Warping (LADTW).

2. SEQUENCE REPRESENTATION OF MUSICAL PIECES

Before evaluating similarity between the query and database music, we should transcribe them into comparable sequences, for example, pitch sequences.

2.1. Extraction of MIDI Theme

The music database consists of MIDI files, which record the note sequence in the format of $(p'_1, d'_1), \ldots, (p'_i, d'_i), \ldots$ where p'_i is the note value and d'_i is the duration. According to the duration d'_i and the frame shift (40 ms in the experiment) of pitch tracking, the note sequence is transformed to one-dimensional pitch sequence $\boldsymbol{p} = (p_1, p_2, \ldots, p_j, \ldots)$, where p_j is the *j-th* pitch value.

2.2. Pitch Tracking of the Query

We employed Simplified Inverse Filter Tracking [15] to extract pitch sequences. In the experiment, the frame length is set to 64 ms and the frame shift is 40 ms. To ensure overall smoothness, the extracted pitch sequence is put through a median filter of order 5. Then the following formula is used to transform them into semitone scale consistent with the format in MIDI files:

$$\text{Semitone} = \log_2(\text{Pitch}/440) * 12 + 69 \tag{1}$$

Finally the query pitch sequence is expressed as $\boldsymbol{q} = (q_1, q_2, \ldots, q_i, \ldots)$, where q_i is the *i-th* pitch value.

3. PREVIOUS WORK

The search process of QBH is to employ an effective algorithm to search the music database for several songs which are most similar to a hummed query. Traditionally, it was regarded as a subsequence matching problem since the query is considered to be part of a song in the database. Two of the most popular algorithms are Linear Scaling (LS) and Dynamic Time Warping (DTW), which are detailed next.

3.1. Linear Scaling

Taking into account that the humming tempo may be inconsistent with that of the target music, LS [3] firstly stretches or compresses the query clip, and then calculates the distance between the query sequence $\boldsymbol{q} = (q_1, q_2, \ldots, q_i, \ldots)$ and song sequence $\boldsymbol{p} = (p_1, p_2, \ldots, p_j, \ldots)$. After trying a variety of possible scaling factors, the optimal scaling factor can result in the least distance. The distance is computed as follows:

$$D(\boldsymbol{q},\boldsymbol{p}) = \frac{\sum_{i=1}^{n} |q_i - p_i|}{n} \tag{2}$$

where $D(\boldsymbol{q},\boldsymbol{p})$ is the distance between sequences $\boldsymbol{q}$ and $\boldsymbol{p}$, q_i is the *i-th* pitch value of $\boldsymbol{q}$, p_i is the *i-th* pitch value of $\boldsymbol{p}$, and n is the total number of query pitches. The algorithm can deal with the problem of different humming tempos. However, if the tempo of a query varies nonlinearly, mismatch will occur due to the fixed scaling factor.

3.2. Dynamic Time Warping

DTW [16], which adopts dynamic programming based distance measure to perform similarity search, can well tackle the issue of nonlinear tempo variation, so it is a more effective melody contour match algorithm [4] [17].

Let $\boldsymbol{q}= (q_1, q_2, \ldots, q_n)$ be a query sequence and $\boldsymbol{p}= (p_1, p_2, \ldots, p_m)$ be a music sequence. To calculate the warping distance, DTW constructs an $n \times m$ distance matrix $D(i,j)_{n\times m}$, where $D(i,j)$ is the distance between sequences $(q_1, q_2, \ldots, q_i)$ and $(p_1, p_2, \ldots, p_j)$, calculated as follows:

$$D(i,j) = \min \begin{cases} D(i-2, j-1) + d(i,j) \\ D(i-1, j-1) + d(i,j) \\ D(i-1, j-2) + d(i,j) \end{cases} \tag{3}$$

And

$$d(i,j) = |q_i - p_j| \tag{4}$$

The boundary conditions for the above recursion can be expressed as:

$$\begin{cases} D(1,j) = d(1,j) & 1 \le j \le m \\ D(i,1) = \infty & 2 \le i \le n \end{cases} \tag{5}$$

After constructing the distance matrix $D(i,j)_{n\times m}$, the least distance between $\boldsymbol{q}$ and $\boldsymbol{p}$ is evaluated as:

$$D(\boldsymbol{q},\boldsymbol{p}) = \underbrace{\min}_{\frac{n}{2} \le j \le m} D(n,j) \tag{6}$$

The two algorithms are designed based on an assumption that the query sequence is a subsequence of the music. Due to humming errors, sometimes only part of the query is a subsequence of the song, so the essential task of QBH is to identify the common subsequence with the largest similarity between the query and music in the database.

4. LOCAL ALIGNMENT FRAMEWORK OF QBH

In Section 3, we have reviewed two popular QBH search algorithms, i.e. LS and DTW. These methods can provide good retrieval results if the query is a subsequence of the song. However, humming errors are inevitable for an unprofessional user, which will badly decrease the retrieval accuracy. According to our experience, users are usually not good at the first few notes, and then they are familiar with the melody. After humming for several seconds, users may forget the melody or lose patience, leading to more humming errors. To verify the above speculation, we divide each query into 5 parts evenly and then employ LS to count the number of wrong pitches between each part and the corresponding music. If the distance between two pitches is bigger than a constant C, the pitch is regarded to be wrongly hummed. Fig. 1 shows the distribution of wrong pitches based on the statistics of 100 queries (C=4) in the experiment. As can be seen, there are more errors in the first and last parts of queries, which is consistent with the intuitive feeling. If an algorithm can tolerate humming errors, especially at the beginning or ending of the clip, it can give a better performance. Based on this consideration, we propose a novel retrieval framework called local alignment, which has been widely used in the field of bioinformatics [18]. This framework aims to identify the common subsequence with the largest similarity between the query and music. By identifying the common subsequence, those humming errors, especially at the beginning or ending of the query, can be discarded, and hence the similarity

score between the query and target song will increase. Local alignment was firstly introduced to QBH by Pardo where he employed note based String Alignment (SA) algorithm to rank the database music [5]. Due to note segmentation errors, note based match algorithms are less effective than pitch based match algorithms [17], so the performance of SA is not satisfactory. What's more, he did not take into account the particularity of the match in QBH systems. To use the local alignment framework more effectively, we implement two algorithms named Local Alignment Linear Scaling (LALS) and Local Alignment Dynamic Time Warping (LADTW).

4.1. Local Alignment Linear Scaling

LS [3] can be used to scale a sequence to match another sequence. To identify the common subsequence between a scaled query $\boldsymbol{q}$ and a song $\boldsymbol{p}$, LALS is designed to compute the local similarity as follows:

$$S(i,i) = \max \begin{cases} S(i-1,i-1) + \frac{s(i,i)+C_{LS}}{n} \\ 0 \end{cases} \tag{7}$$

And

$$s(i,i) = -|q_i - p_j| \tag{8}$$

where $S(i,i)$ is the similarity between sequences $(q_1, q_2, \ldots, q_i)$ and $(p_1, p_2, \ldots, p_i)$, $s(i,i)$ is the similarity of q_i and p_i , n is the length of the sequences and C_{LS} is a preset reward value. It should be noted that $s(i,i)$ is non-positive, which equals to the negative of the distance. C_{LS} is a compensation value of $s(i,i)$. It ensures that small distance brings positive similarity and big distance brings negative similarity. In Eq. (7), by comparing the similarity with 0, LALS can reduce the negative impact of mismatch fragments that have negative similarities at the beginning of the query. Then we can apply the following formula to obtain the maximum similarity:

$$S(\boldsymbol{q},\boldsymbol{p}) = \underbrace{\max}_{1 \le i \le n} S(i,i) \tag{9}$$

By finding the maximum similarity from $S(i,i)$ $(1 \le i \le n)$ instead of $S(n,n)$, LALS can abandon errors at the ending of the query. As these humming errors are not taken into account for similarity calculation, the target song can obtain a higher score.

4.2. Local Alignment Dynamic Time Warping

In the task of QBH, due to the arbitrary and variable humming tempo, the measure of warping distance is more accurate than that of linear distance. LADTW is a fusion of DTW and local alignment for melody contour match [17], so it can calculate the common subsequence with the largest warping similarity between two sequences. If the query pitch sequence $\boldsymbol{q}$ is $(q_1, q_2, \ldots, q_n)$ and the music pitch sequence $\boldsymbol{p}$ is $(p_1, p_2, \ldots, p_m)$, then we construct an $n \times m$ similarity

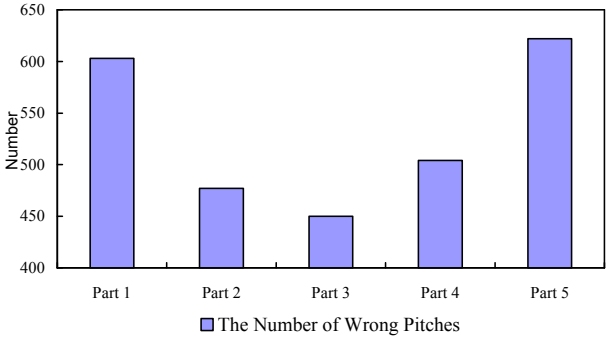

Fig. 1. The total number of wrong pitches in different parts of queries according to the statistics of 100 queries which are randomly selected from the experimental dataset. In the statistics, each query is evenly split into 5 parts and then each part is compared with the corresponding music by using LS, respectively.

matrix $S(i,j)_{n \times m}$, where $S(i,j)$ is the similarity between sequences $(q_1, q_2, \ldots, q_i)$ and $(p_1, p_2, \ldots, p_j)$. After that, we introduce the following recursion function to compute the similarity:

$$S(i,j) = \max \begin{cases} S(i-2,j-1) + s(i,j) + wC_{DTW} \\ S(i-1,j-1) + s(i,j) + C_{DTW} \\ S(i-1,j-2) + s(i,j) + wC_{DTW} \\ 0 \end{cases} \tag{10}$$

And

$$s(i,j) = -|q_i - p_j| \tag{11}$$

In Eq. (10), $S(i,j)$ is the similarity between sequences $(q_1, q_2, \ldots, q_i)$ and $(p_1, p_2, \ldots, p_j)$, $s(i,j)$ is the similarity of the *i-th* pitch of $\boldsymbol{q}$ and the *j-th* pitch of $\boldsymbol{p}$ as shown in Eq. (11), C_{DTW} is a preset reward value based on a priori statistic, whose function is to compensate for the similarity $s(i,j)$ and bias the search toward a longer match, and w is the weight in different paths. Local alignment is introduced by comparing the similarity with 0 as shown in Eq. (10).

The boundary conditions for the LADTW recursion are shown as follows:

$$\begin{cases} S(i,1) = \max \begin{cases} s(i,1) + C_{DTW} \\ 0 \end{cases} , 1 \le i \le n \\ S(1,j) = \max \begin{cases} s(1,j) + C_{DTW} \\ 0 \end{cases} , 2 \le j \le m \end{cases} \tag{12}$$

The above equations indicate that the optimal path can start from anywhere of the two sequences. After computing the similarity matrix $S(i,j)_{n \times m}$, we can easily obtain the maximum similarity by:

$$S(\boldsymbol{q},\boldsymbol{p}) = \underbrace{\max}_{1 \le i \le n, 1 \le j \le m} S(i,j) \tag{13}$$

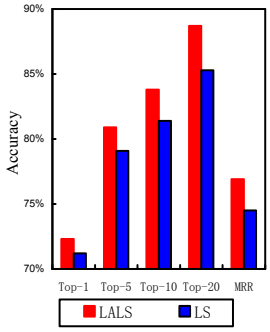

Fig. 2. Retrieval accuracy of LS and LALS.

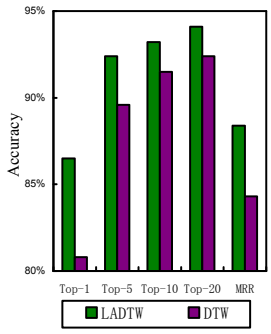

Fig. 3. Retrieval accuracy of DTW and LADTW.

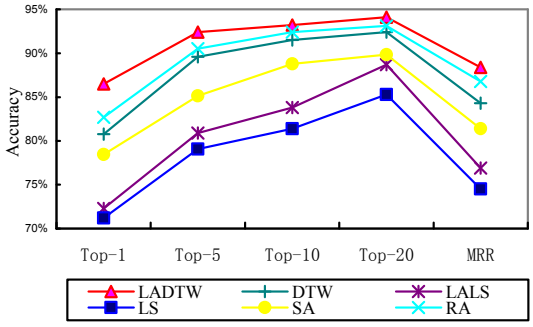

Fig. 4. The performance of different algorithms.

Similar to LALS, the introduction of 0 in Eq. (10) can eliminate the impact of serious errors, especially at the beginning of queries. By using Eq. (13) to find the maximum similarity from the entire similarity matrix instead of the last row of the matrix (as shown by Eq. (6)), the algorithm is robust to errors at the end of queries.

5. EXPERIMENTS

5.1. Experimental Dataset

The music dataset in the experiments is the MIREX 2010 QBH Think IT corpus [19]. The corpus consisted of 355 queries and 106 MIDI files. All of the queries were hummed from anywhere. We added 5,000 noise MIDI files, crawled from the Internet, to form a large database. The evaluation measurements are mean reciprocal rank (MRR) and Top-K hit rate. MRR is the average of the reciprocal ranks across all queries, calculated by $MRR = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{rank_i}$, where n is the number of queries and $rank_i$ is the ranking of the correct song for each query. The Top-K hit rate is the percentage that the correct song is ranked within the top K of all returns.

5.2. Experiments of LALS

Fig. 2 shows experimental results of LALS and LS. The empirical value of C_{LS} in Eq. (7) was set to 3. As can be seen, LALS improved the retrieval accuracy compared with LS. The accuracy of Top-20 increased most with the relative improvement up to 23.1%. Due to the limitation of LS, which can not deal with the nonlinear variation of humming tempos, the retrieval accuracy of LALS is still not high despite the use of local alignment.

5.3. Experiments of LADTW

Fig. 3 shows the performance of LADTW and DTW. In the experiments, C_{DTW} in Eq. (10) was empirically set to 0.8, and w was set to 1.6. As can be seen, the retrieval accuracy of LADTW was much higher than DTW due to the application of local alignment. The relative improvement of Top-1 was up to 29.7% and that of MRR was up to 26.1%. Since DTW itself has already reached a good performance, the performance of LADTW is much higher by using the local alignment.

5.4. Experiments of Different Algorithms

Fig. 4 shows the performance of several different algorithms, such as LADTW, DTW, LALS, LS, RA [13] and String Alignment (SA) [5]. As can be seen from Fig. 4, despite the use of local alignment, the performance of SA and LALS was still very low, because note based SA is much worse than pitch based algorithms and LALS can not deal with the problem of nonlinear tempos. In contrast, the retrieval accuracy of LADTW was highest compared with all other algorithms owning to the application of local alignment and the baseline high performance of DTW algorithm.

6. RELATION TO PRIOR WORK

In recent years, the retrieval accuracy of QBH has reached a bottleneck, so most researches focused on the speed improvement [11] [12] [13] [20] or fusion of different methods [7] [21]. Few papers were published on the improvement of a single match algorithm in the past two years. In this paper, we proposed a framework of local alignment for QBH. It was applied to two popular algorithms, i.e. LS [3] and DTW [17]. The new algorithms improved the performance greatly.

7. ACKNOWLEDGEMENTS

This work was partially supported by National Natural Science Foundation of China under Grant No.61273217 and 61171193, the Next-Generation Broadband Wireless Mobile Communications Network Technology Key Project under Grant No. 2011ZX03002-005-01, the 111 project under Grant No.B08004, and Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

8. REFERENCES

[1] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668 –696, april 2008.

[2] A. Ghias, J. Logan, D. Chamberlin, and B. Smith, "Query by humming: musical information retrieval in an audio database," in *Proceedings of the third ACM international conference on Multimedia*. ACM, 1995, pp. 231–236.

[3] J. Jang, H. Lee, and M. Kao, "Content-based music retrieval using linear scaling and branch-and-bound tree search," in *IEEE International Conference on Multimedia and Expo, Waseda University, Tokyo, Japan*, 2001.

[4] J. Jang and H. Lee, "Hierarchical filtering method for content-based music retrieval via acoustic input," in *Proceedings of the ninth ACM international conference on Multimedia*. ACM, 2001, pp. 401–410.

[5] B. Pardo, J. Shifrin, and W. Birmingham, "Name that tune: A pilot study in finding a melody from a sung query," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 4, pp. 283–300, 2003.

[6] Q. Wang, Z. Guo, B. Li, G. Liu, and J. Guo, "Tempo variation based multilayer filters for query by humming," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3034–3037.

[7] Z. Guo, Q. Wang, G. Liu, J. Guo, and Y. Lu, "A music retrieval system using melody and lyric," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, july 2012, pp. 343 –348.

[8] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*, ser. SCG '04. New York, NY, USA: ACM, 2004, pp. 253–262.

[9] Q. Wang, Z. Guo, G. Liu, and J. Guo, "Entropy based locality sensitive hashing," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, march 2012, pp. 1045 –1048.

[10] Q. Wang, Z. Guo, and G. Liu, "Boundary-expanding locality sensitive hashing," in *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*. IEEE, 2012, pp. 358–362.

[11] M. Ryynanen and A. Klapuri, "Query by humming of midi and audio using locality sensitive hashing," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 31 2008-April 2008, pp. 2249 –2252.

[12] Y. Yu, M. Crucianu, V. Oria, and E. Damiani, "Combining multi-probe histogram and order-statistics based LSH for scalable audio content retrieval," in *Proceedings of the international conference on Multimedia*, ser. MM '10. ACM, 2010, pp. 381–390.

[13] Z. Guo, Q. Wang, J. Guo, and G. Liu, "A query by humming system based on locality sensitive hashing indexes," *Signal Processing*, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/-S016516841200326X

[14] Q. Wang, Z. Guo, G. Liu, J. Guo, and Y. Lu, "Query by humming by using locality sensitive hashing based on combination of pitch and note," in *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, july 2012, pp. 302 –307.

[15] J. Markel, "The SIFT algorithm for fundamental frequency estimation," *Audio and Electroacoustics, IEEE Transactions on*, vol. 20, no. 5, pp. 367–377, 1972.

[16] J. Kruskall and M. Liberman, "The symmetric time warping algorithm: From continuous to discrete," *Time Warps*, 1983.

[17] R. Dannenberg, W. Birmingham, B. Pardo, N. Hu, C. Meek, and G. Tzanetakis, "A comparative evaluation of search techniques for query-by-humming using the musart testbed," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 5, pp. 687–701, 2007.

[18] T. Smith and M. Waterman, "Identification of common molecular subsequences," *Molecular Biology*, vol. 147, pp. 195–197, 1981.

[19] Music information retrieval evaluation exchange (MIREX). [Online]. Available: http://www.music-ir.org/mirex/wiki/2010:Query_by_Singing/Humming

[20] W.-H. Tsai and Y.-M. Tu, "An efficient query-by-singing/humming system based on fast fourier transforms of note sequences," in *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, july 2012, pp. 521 –525.

[21] Z. Guo, Q. Wang, L. Yin, G. Liu, and J. Guo, "Query by humming via hierarchical filters," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 3021–3024.