Unified Algorithm for Melodic Music Similarity and Retrieval in Query by Humming

Velankar Makarand and Kulkarni Parag

Abstract Query by humming (QBH) is an active research area since a decade with limited commercial success. Challenges include partial imperfect queries from users, query representation and matching, fast, and accurate generation of results. Our work focus is on query presentation and matching algorithms to reduce the effective computational time and improve accuracy. We have proposed a unified algorithm for measuring melodic music similarity in QBH. It involves two different approaches for similarity measurement. They are novel mode normalized frequency algorithm using edit distance and n-gram precomputed inverted index method. This proposed algorithm is based on the study of melody representation in the form of note string and user query variations. Queries from four non-singers with no formal training of singing are used for initial testing. The preliminary results with 60 queries for 50 songs database are encouraging for the further research.

Keywords QBH · Music similarity · Pattern matching · Information retrieval

1 Introduction

Query by humming (QBH) is one of the most natural ways of expressing query and search a song from the musical database. Researchers in this domain of computational musicology are working on QBH since last decade or so. Content-based retrieval with the melody of the song as input with QBH is a challenge and has very limited success in commercial applications so far. The interfaces for song search used and available today are mainly based on keywords on metadata associated with the song.

V. Makarand (✉)
Cummins College of Engineering, Pune, India
e-mail: makarand.velankar@cumminscollege.in

K. Parag
Iknowlation Research Labs Pvt. Ltd, Pune, India
e-mail: paragindia@gmail.com

S. Bhalla et al. (eds.), *Intelligent Computing and Information and Communication*,
Advances in Intelligent Systems and Computing 673,
https://doi.org/10.1007/978-981-10-7245-1_37

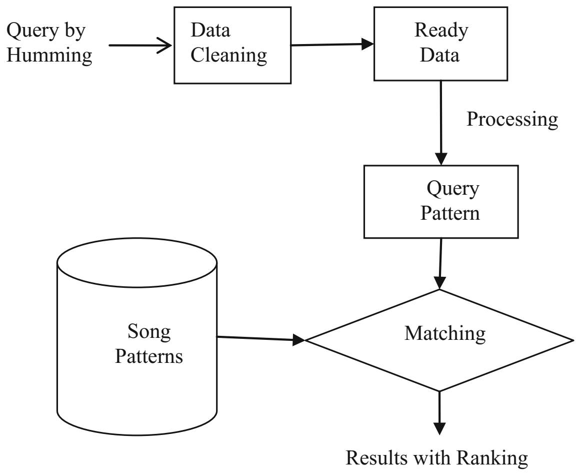
A typical query by humming system can be represented with block diagram as shown in Fig. 1. Users submit the query in the form of hummed query. At data cleaning stage, different signal processing techniques are applied, such as background noise removal, melody segmentation, etc. to make query ready for further processing. The query is processed to identify and represent an underlying pattern which is matched with song patterns already stored in the database to find best matches. Depending on the best possible matches results are displayed with ranking.

One of the major challenges in QBH is different users have different tonic or fundamental frequency and identifying tonic becomes an initial challenge. The query can be presented by the same user using different tonic at different instances and at a different tempo. The dynamic time warping algorithm is used for normalizing queries on timescale. Query transposes are used for same query representation considering possible tonic information. This leads to many possible representations of the same query and increases the query processing time. Since queries are likely to be run for millions of songs in the commercial applications, acceptable time complexity, and results are necessary for any commercial application.

A novel mode normalized frequency (MNF) algorithm for edit distance measure is proposed which eliminates the need of query transposes. N-gram inverted index method uses precomputed n-grams of songs. Inverted index of each n-gram includes a list of songs having these patterns. It is useful for the efficient matching of the identified n-grams from the QBH. The performed experiment uses the raw query data and presented results for different algorithms and unified algorithm. The novel unified algorithm narrow downs the possible results using n-gram approach and then apply MNF algorithm for edit distance on potential shortlisted songs.

The paper is organized in the following manner. Section 2 gives necessary musical background and considerations used in the computation. Section 3 refers to

Fig. 1 Query by humming system



related work, methodologies, and different approaches used by the researchers. Section 4 detailed basic framework and approximate pattern matching algorithms used for the study. Section 5 explains computational approach using MNF algorithm. Section 6 covers details of results obtained using different algorithms. The contribution of the paper is summarized with future directions in Sect. 7.

2 Music Background and Contemplations

Musical notations are represented on the logarithmic scale. The sample octave is represented with 12 notes having frequencies F_1 to F_{12}. The equal tempered scale is one of the most widely used musical scales. General formula for the calculation of frequencies F_k from base reference frequency F_1 for an octave is

$$\begin{aligned} F_k &= F_1 \cdot 2^{k/12} \quad \text{where } k = 2 \text{ to } 12. \\ F_{12} &= F_1 \cdot 2^{12/12} = F_1 \cdot 2 = 2\,F_1 \end{aligned} \tag{2.1}$$

The 12th note as F_{12} has a frequency double of F_1 and it marks the end of the current octave and beginning of next octave. For processing of the audio query, 100 samples are considered per second. In order to perceive any note, general minimum duration of note to be sung is about 30–40 ms. Observations of the query by untrained singers and general song patterns revealed 100 ms as the minimum duration of any note as a rough estimate., i.e., 10 consecutive samples of the same frequency are required to decode a specific note. These estimate found to be appropriate for QBH.

Notes sung during query by users are not likely to be perfect as singers. They are likely to deviate from the original note frequency. For such imperfect queries to accommodate possible errors in the query, the algorithm design has considered a range of frequency keeping actual frequency as a center to represent the specific note. It has covered entire frequency spectrum using the range such that each frequency will be associated with some or the other note. Another important observation is users can have different pitch range and it reflects in the queries with different frequencies for same song rendition by different users. The same person can also submit the query in different scales and octaves. This leads to the concept of frequency transpose considerations. As note sequence, F_1 F_3 F_2 will be perceived similar to all its aliases as F_2 F_4 F_3 or F_9 F_{11} F_{10}, etc. These sequences represent same relative difference as +2, −1 in subsequent notes as compare to previous note. Relative differences are used for precalculation of n-grams for songs which eliminates the need of any transposes according to the tonic.

3 Related Work

Research on QBH has been prevailing since last two decades with many researchers attempting to get better results. These results are influenced by two major factors as quality of query itself and the algorithms used for search and comparison. Algorithms not only need to be efficient in terms of accuracy but also needs to have better time complexity considering millions of songs to be searched for the query in the application. Queries are represented in the form of a string of notes in the majority of the work with some exceptions as additional information as rhythm or duration of notes with a pitch. Results are evaluated as top 10 hit ratios or mean reciprocal rank. Top 10 hit ratios give a percentage of queries for which intended song appears in top 10. Mean reciprocal rank (MRR) is calculated using the following formula. In case of four queries for the algorithm, the intended song appear at say position 1, 5, 2, and 10, then MRR will be (1/1 + 1/5 + 1/2 + 1/10)/4 = 0.45. MRR value is having a range between 0 and 1 with more value indicating better results and lower value represents poor results.

Some standard datasets are available for western music but as per our knowledge, no standard dataset is available for the queries and Hindi songs to compare algorithms and test accuracies. Tansen [1] QBH for Hindi songs tested queries using different syllables, such as “Ta”, “La”, “Da”, and evaluated the impact of different syllable on the performance of the system. Tonal representation of melody with semitone and octave abstraction along with Q max algorithm is used to compute similarity [2]. Authors emphasized the need of query self-tuning to improve results considering the results for the subjects with good tuning the top 10 hit ratio was 64%. Existing QBH systems have slow searching speeds and lack practical applications and proposed GPU-accelerated algorithm with parallel processing to improve search efficiency by 10 times [3]. Considering wide use of mobile for QBH, the query needs to be small in duration and testing of commercial applications revealed the time required is a crucial factor [4].

Importance of tracking fundamental frequency and its impact on the performance of the system is studied and proposed further research for deeper insight into different methods [5]. Different approaches for melodic similarity for Indian art music are evaluated and it is observed that results are sensitive to distance measures used by the algorithms [6]. The results are not comparable, as in the majority of the cases datasets used are different by the different researchers.

Recently, new approaches have been applied to improve the accuracy of QBH system by researchers. In addition to the conventional method, the results are ranked according to user preferences or history to identify user taste and the songs most likely queried at top positions [7]. Use of combining lyrics information available with the user in melody achieved 51.19% error reduction for the top 10 results [8]. Segment-based melody retrieval approach can solve the problem of noise and tempo variation in the query better than global linear scaling [9]. Despite extensive research on the QBH, present systems for song search mainly relies on keyword-based search with limited use in the commercial applications. This is our main motivation for the research work in QBH domain of computational musicology. It has revealed many further challenges and opportunities for the research with various directions.

4 Basic Framework and Pattern Matching Algorithms

The humming query is processed using open-source tool PRAAT for experiments. The default settings of PRAAT are used for query data processing [10]. Framework for content-based music retrieval is proposed using human perception [11] and this work fits as a module in it. The sampling rate for pitch samples used is 100 samples per second. No data cleaning signal processing methods used for our experimentation, however, removal of noise can improve accuracies further.

Pitch listing information with samples is further converted to note string with the novel approach Mode Frequency Normalization. This algorithm is the outcome of study and experimentation with queries and song patterns. Approximate pattern matching algorithms are necessary to match the humming query patterns as exact matches are very unlikely for the non-singers queries. Observations about hummed queries revealed that some queries are time stretched, whereas some of them are compressed on the timescale. In some cases, few notes are prolonged as compared to others. It prompted us to consider compressed queries on the timeline with removing duplicate notes in succession.

For query pattern matching, edit distance and n-gram methods are used. Use of Euclidian distance and n-grams for pattern matching on compressed and non-compressed queries generated four possible alternatives as (1) Simple Euclidian distance. (2) Compressed Euclidian distance. (3) N-gram method. (4) Compressed n-gram method. For compressed queries, the original song patterns are also compressed for calculating distance measure. The cost of calculation for compressed queries is less but it eliminates the comparison of pattern with same successive notes.

Edit distance method uses Euclidian distance measure with distance as a minimum number of insertions, deletions, or replacements required to match two strings. The less the edit distance more similar the song will be for query. Dynamic programming approach is used to find edit distance. For n-gram approach, the calculations are restricted till 4-grams as it was observed unnecessary to calculate further grams as occasionally matches are found for higher grams. In case of 2-gram approach, all note patterns with two successive notes with relative distance are identified from the query and compared with song patterns to an identified number of occurrences of 2-grams. 2-gram approach is like substring matching algorithm. The more number of n-gram matches indicate more likely the pattern matching for the given query.

The inverted index structure is used with pre-calculated relative pitch 2-grams (RP2G), 3-grams (RP3G) and 4-grams (RP4G) from the song patterns. This approach eliminates the need for substring matching of the query with each song pattern. To find matching n-grams of the query, pre-calculated values of n-grams are used. For query example string with relative note distances as +1 +1 +4, the 2-grams are +1 and +4. An inverted index for string +1 will contain a list of songs containing string +1 with in sorted order of a number of occurrences of +1. For each song, 2-gram measure will be the sum of all matching 2-grams from the query. The more the value of 2-grams, the more the song is likely to similar to the query. Similarly 3-grams (+1 +1 and +1 +3 in the above example) and 4-grams (+1 +1 +4 in the above example) are also pre-calculated for song list. Results are evaluated for different pattern matching algorithms using n-grams and edit distance methods.

The unified algorithm uses four individual algorithms in combination as per the need. It advances to the next algorithm if desired results are not obtained in the previous algorithm. At first, it uses RP4G then RP3G and RP2G followed by MNF algorithm. It uses minimum cutoff value for n-grams. Present cutoff values are decided considering our observations. Cutoff is minimum grams matched of the query with the song patterns. Cutoff values considered are 2, 5, and 8, respectively, for RP4G, RP3G, and RP2G at present, however, further fine tuning is possible. The implementation of individual algorithms and unified algorithms is done using C programming language and the results are tested for queries from different users. The proposal is to submit the tested code to PRAAT open-source tool at the later stage and make it available to all.

5 Mode Normalized Frequency (MNF) Algorithm

The note patterns for about 50 Hindi Popular songs studied which are selected randomly from a different era. It was observed that few notes are repeated in the majority of the note sequence pattern. These notes are different in different song patterns, however, the trend prevails. Further introspection revealed that one note is appearing more number of times compared to others. These observations lead us to consider mode concept in statistics which is maximum occurrence symbol or value in the set. Assuming this note as a reference, the song sequences are normalized and converted them to mode normalized frequency form.

For normalized notes pattern, notes are represented using A–Z alphabetic sequence with N as middle alphabet considered as a reference. All notes are considered relative to it, as a note with distance +2 as P or −2 as L with respect to N and so on. The original song notes pattern is converted to the normalized pattern by keeping a relative distance of notes same.

MNF algorithm for query processing:

1. Read pitch listing for the Query
2. Find maximum occurring frequency in the pitch listing (N)
3. Find other notes and their frequencies using the formula 2.1
4. Find range of each note to cover entire frequency range

5. Convert pitch listing of Query to notes representation
6. Read converted QBH for notes
7. If note appears 10 times in a consecutive manner add to output sequence
8. Continue reading the sequence till end
9. Output mode normalized sequence.

MNF algorithm eliminates the need for computation for transposes in Euclidian distance measure; however, some exceptions for songs or queries do need transpose calculations considering the pattern with non-repeating note for better results. Following sample for song and query with appropriate representations gives an idea about the conversion and representations used. * followed by note represents next octave in the notations. Sa* appearing having more occurrences in compressed note pattern is replaced by N in the normalized query pattern as shown in the example below. Pattern matching is done using edit distance method in MNF algorithm.

Compressed song note sequence pattern: g Sa* ni Sa* re* ni Dh ni Sa* Dh dh Dh ni
Relative note sequence pattern: +8 −1 +1 +2 −3 −2 +2 +1 −3 −1 +1 +2
Normalized Query pattern for MNF edit distance algorithm: FNMNPMKMNKJKM

6 Analysis of Results

Our experiments include humming samples recorded using ordinary recording mike of laptop and mobile devices. High-quality recording system was not used purposely as it would not be available with common listeners to generate queries. For the humming queries, the samples are generated from people with no formal training of singing. This has helped us to test our algorithms for imperfect samples from the unprofessional users. In the experiments, 60 humming queries are used with a rendition by four amateur youngsters. The queries are supposed to retrieve one of the songs from the list of 50 sample songs as a part of the top songs of the ranked list.

Analysis of results is done for compressed queries and compressed song patterns using RP4G, RP3G, RP2G, MNF algorithm, and Unified approach. Present experiments show unified algorithm is promising as it is time efficient for the majority of queries to reduce the average computational time and overall accuracy is also improved. It was observed in the run stage of a unified algorithm that out of 60 queries, for 24 queries only 4-grams got results, for 21 queries 3-grams got results, for 8 queries 2G used and for 7 queries need to be occurred to use MNF algorithm for top 10 results.

Individual user query and overall results for top 3 hit ratios are shown in Table 1 for all algorithms. Considering the mobile query interface, top 3 results carry more importance than top 10, thus top 3 results are shown. It can be observed from the table that for individual algorithms, RP3G is better compare to rest followed by

Table 1 Top 3 hit ratios in percentage

Queries	RP4G	RP3G	RP2G	MNF	Unified
User 1	54.5	81.8	43.4	54.5	81.8
User 2	36.4	63.6	30.0	60.0	81.8
User 3	0	53.3	20.1	46.1	46.6
User 4	30.8	79.9	36.1	46.1	76.9
Overall	30.4	68.9	33.9	51.7	71.9

Table 2 MRR for different algorithms

Algorithms	RP4G	RP3G	RP2G	MNF	Unified
MRR	0.2473	0.5076	0.232	0.45	0.59

MNF algorithm. The proposed unified algorithm has the best accuracy out of all algorithms. Another observation is that results vary according to the user's query quality. It can be seen from the table that user 3 generated the poor-quality queries among all others.

The mean reciprocal ratio or MRR gives an accuracy of methods. More the MRR value, better the algorithm. Table 2 shows MRRs for different algorithms. The unified algorithm has obtained better accuracy as compared to others for the test data used. These preliminary results are encouraging for further experimentations.

7 Summary and Future Directions

The unified algorithm a novel approach proposed in this paper is promising as it reduces the cost of computation time with precomputed the inverted index for fast look up. For 75% of the queries, the results obtained using RP4G and RP3G in the unified algorithm and for the rest of the queries, RP2G and MNF algorithm were used. The initial results show that the unified algorithm is time efficient considering the average time and the results are also improved with MRR as 0.59 as compared to 0.5076 of the best individual algorithm of RP3G and 0.45 of the developed MNF algorithm.

Since users normally refer to the first page of results, if the intended song appears quite below in the list it is not likely to be found. Thus, top 3/5/10 hit ratios become more important than MRR from user's point of view, however, MRR is a better major for the accuracy of the algorithm. Further experimentation with more queries from different users will help to fine-tune the unified algorithm for better accuracy. Query and song segmentation approach needs to be evaluated to see further precision. This will help us to fine-tune the unified algorithm and build a robust system.

Acknowledgements The authors gratefully acknowledge the support by MKSSS's Cummins College of Engineering for providing experimental setup and the efforts by our UG students Aditi Pawle, Snehal Jain, Sonal Gawande, and Sonal Avhad for the active help in preparing query samples and experiments. Volunteer singer's contribution for generating queries is highly appreciable. We would like to thank Dr. Sahasrabuddhe H. V. for his valuable suggestions and inputs related to musical knowledge and experiments.

References

1. Kumar P, Joshi M, Hariharan S, Rao P: Sung note segmentation for a query-by-humming system. Intl Joint Conferences on Artificial Intelligence IJCAI (2007).
2. Salamon J, Serra J, Gómez E: Tonal representations for music retrieval: from version identification to query-by-humming. International Journal of Multimedia Information Retrieval. 2(1) pp 45–58, (2013).
3. Ruan L, Wang L, Xiao L, Zhu M, Wu Y: A Query-by-Humming System based on Marsyas Framework and GPU Acceleration Algorithms. Appl. Math. pp 261–72, Feb (2013).
4. Chandrasekhar V, Sharifi M, Ross DA: Survey and Evaluation of Audio Fingerprinting Schemes for Mobile Query-by-Example Applications. ISMIR Vol. 20, pp. 801–806 (2011).
5. Molina E, Tardón LJ, Barbancho I, Barbancho AM: The Importance of F0 Tracking in Query-by-singing-humming. In ISMIR pp. 277–282, Nov (2014).
6. Gulati S, Serra J, Serra X.: An evaluation of methodologies for melodic similarity in audio recordings of indian art music. In Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference pp. 678–682 Apr, (2015).
7. Liu NH: Effective Results Ranking for Mobile Query by Singing/Humming Using a Hybrid Recommendation Mechanism. IEEE Transactions on Multimedia. pp 1407–20, (2014).
8. Wang CC, Jang JS: Improving query-by-singing/humming by combining melody and lyric information. IEEE/ACM Transactions on Audio, Speech, and Language Processing. pp 798–806 (2015).
9. Liu, N. H: Effective Results Ranking for Mobile Query by Singing/Humming Using a Hybrid Recommendation Mechanism. IEEE Transactions on Multimedia, 1407–1420(2014).
10. Ramesh V: Exploring Data Analysis in music using tool praat. ICETET. IEEE International Conference, pp. 508–509, (2008).
11. Makarand, Velankar, and H. V. Sahasrabuddhe: Novel Approach for Music Search Using Music Contents and Human Perception, IEEE International Conference on Electronic Systems, Signal Processing and Computing Technologies ICESC, (2014).