# EFFECTIVENESS OF N-GRAM FAST MATCH FOR QUERY-BY-HUMMING SYSTEMS

*Jue Hou*[1], *Dan-ning Jiang*[2], *Wen-xiao Cao*[1], *Yong Qin*[2], *Thomas Fang Zheng*[1], *Yi Liu*[1]

[1]Center for Speech and Language Technologies, Division of Technology Innovation and Development,
Tsinghua National Laboratory for Information Science and Technology;
Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]IBM China Research Lab, Beijing, China
{houj, caowx}@cslt.riit.tsinghua.edu.cn, {jiangdn, qinyong}@cn.ibm.com, {fzheng, eeyliu}@tsinghua.edu.cn

## ABSTRACT

To achieve a good balance between matching accuracy and computation efficiency is a key challenge for Query-by-Humming (QBH) system. In this paper, we propose an approach of n-gram based fast match. Our n-gram method uses a robust statistical note transcription as well as error compensation method based on the analysis of frequent transcription errors. The effectiveness of our approach has been evaluated on a relatively large melody database with 5223 melodies. The experimental results show that when the searching space was reduced to only 10% of the whole size, 90% of the target melodies were preserved in the candidates, and 88% of the match accuracy of system was kept. Meanwhile, no obvious additional computation was applied.

***Index Terms***— N-gram, Fast match, Query-by-Humming

## 1. INTRODUCTION

The Query-by-Humming (QBH) system allows users to retrieve songs by singing or humming a fragment of the melody. The singing or humming interface is regarded as the most natural way for song retrieval, and it is particularly important in the cases that people cannot remember any key words of a song.

A variety of techniques have been developed to improve the performance of QBH systems in recent years. Most of the earlier systems were note-based, in which the match was performed on the note sequences of the query and the melody [1]. The major drawback of such systems is that note transcription of the query is often erroneous, which leads to the system performance degradation. To solve this problem, later systems omitted note transcription and directly matched on the frame-based pitch contour [2, 3, 4]. It is shown that the frame-based system is able to achieve better performance with the increased computation cost.

Recently, a multi-stage matching scheme was proposed with the goal of achieving a good balance between recognition accuracy and computational cost [4]. In this approach, the candidates are first pruned by one or more layers using fast match with a high recalling rate and the remaining candidates are rescored with the most accurate but computation-intensive algorithm.

For practical QBH systems, fast match is a key issue to guarantee good performance of melody retrieval with acceptable computation cost. However, to the best of our knowledge, the relevant research is limited, especially for large-scale melody database.

Previous studies showed positive results of using n-gram as a fast match method in melody match systems when the query is input in a symbolic format of music via keyboard [5]. However, when the query is the acoustic signal of singing or humming, which means note transcription is needed before the n-gram match, the performance degraded much because of the transcription errors. As described in [6], if the n-gram match returned 10% of the database for the next stage of accurate match, about 50% to 60% of the correct results were lost. Meanwhile, in order to keep high percentage to guarantee sufficient correct candidates, more computation time is required.. In this case, the obtained accuracy and computational time cost are unacceptable for a real QBH system.

In this paper, we propose an approach of n-gram based fast match. We first recognize the query signal into a note sequence by applying a robust statistical method [7]. We then compensate the recognition errors by fuzzy rules that can be designed based on the analysis of frequent transcription errors. As a result, the mismatch of n-gram caused by transcription errors can be reduced. The experiments performed on a relatively large database have shown that the n-gram fast match had a satisfactory performance. About 90% of the target melodies were preserved when the searching space was cut down to 10% of the whole size.

The paper is organized as follows. Section 2 describes the n-gram fast match algorithm. Section 3 presents a prototype system of QBH to evaluate the n-gram algorithm. The experiment results are reported in section 4. We concluded in Section 5.

## 2. N-GRAM FAST MATCH ALGORITHM

In general QBH system, the query signal is first transcribed into a sequence of notes $Q=(q_1,q_2,\cdots,q_m)$, and each note is represented as a pair of tone and duration $q_i=(t(q_i),d(q_i))$ ($0<i\le m$). The melody database contains a list of melodies $(C_1,C_2,\cdots,C_K)$, and each item is also represented as a sequence of notes $C_k=(c_1^k,c_2^k,\cdots,c_{n(k)}^k)$ ($0<k\le K$). The fast match algorithm needs to return a relatively small number of match candidates, further pass them to the accurate match stage to get the final results. Each match candidate is represented as $(q_i,C_j^k)$, showing that $q_i$ should be matched with $C_j^k$ in the k-th melody. $q_i$ is not necessary to be the first note in the query, and if it is in the middle, we'll perform the accurate match both forwardly and backwardly

978-1-4244-4291-1/09/$25.00 ©2009 IEEE

for the segment after and before the matched pair, respectively. Then sum the scores of these two parts to get the final match score.

In general, n-gram fast match algorithm locates probable match candidates by checking the appearance of the same n-grams extracted from the queries in the melody database. Melodies in the database are first broken into overlapping n-grams and a reverse index mapping the n-gram to a set of match entries is built. Then, the query note sequence is also broken into overlapping n-grams. Due to transcription errors, the correct n-gram can be lost in the query note sequence, further degrades the fast match performance. Thus, we propose a novel error compensation method to solve this problem. Candidates for the accurate match stage are generated by querying the index with the n-grams, and they are pruned before being passed to the accurate match.

## 2.1. N-gram extraction

In n-gram extraction process, we only consider the tone of a note. The note sequence is first pre-processed by combining adjacent notes with the same tone. To eliminate the effect of different key scales, the n-gram is represented as a vector of tone intervals. For example, a note sequence labeled with the MIDI tone as follows:

60, 67, 69, 67, 65, 64, 62, 60

The overlapping 4-grams are:

<0, +7, +2, -2>
<0, +2, -2, -2>
<0, -2, -2, -1>
<0, -2, -1, -2>
<0, -1, -2, -2>

The interval represents the difference between two adjacent notes in tone, so the first values in the n-grams are always 0. The n-grams extracted from the database are then used to build the reverse index, while those extracted from the query need to be further processed to compensate the transcription errors in order to get higher match accuracy.

## 2.2. Transcription error compensation

It is shown in [6] that the note transcription errors lead to severe mismatches in n-gram, further degrades the performance. To solve this problem, previous studies used coarse quantization, which quantized the note intervals into a smaller number of ranges (e.g. in [6], seven ranges were used: <-7, -7 to -3, -2 to -1, unison, 1 to 2, 3 to 7, >7). However, the performance is still not satisfactory because coarse quantization also includes many irrelevant n-grams. Intuitively, a good error compensation method should minimize the number of mismatches as well as irrelevant n-grams.

After analyzed the most frequent errors in the transcription results, we designed a set of fuzzy rules to expand the n-grams extracted from the query. The expanded set of n-grams is more likely to include the correct n-gram with a relatively small size. We observed two types of most frequent errors in the transcription results:

*Fine tone error*: the transcribed tone is 1-semitone higher or lower than the actual tone. Figure 1(a) shows one example, where the upper line shows the actual sequence and the lower line shows the transcribed sequence. It is clear that the third note is recognized incorrectly.

*"Sliding" insertion error*: when the pitch contour "slides" from a lower tone to a higher tone, a short insertion error often occurs in the "sliding" part between the two notes. The tone of the inserted note is usually in the middle of the two notes. Figure 1(b) shows one example, the second note of the transcribed sequence is such an insertion error.

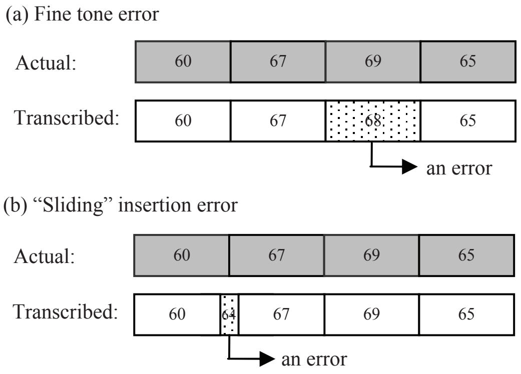

Figure 1. Examples of frequently happened transcription errors: (a) fine tone error; (b) "sliding" insertion error.

Due to the effect of the above two errors, mismatches of n-grams are likely to occur. Hence, two fuzzy rules were designed in order to expand more intervals besides those directly derived from the transcribed note sequence:

*Rule1*: For each two adjacent notes, the intervals with 1-semitone higher or lower than the original interval are added.

*Rule2*: If a note is very short (e.g., shorter than 0.1 s) and its tone is higher than its previous note and lower than its next note, then an interval between its previous note and next tone is added.

Figure 2 illustrates the expanded graph of note intervals for the transcribed note sequence shown in Figure1 (b). Each node in the graph denotes a transcribed note, and each arc between two nodes represents a tone interval. A real line arc shows that the associated interval is directly derived from the transcribed note sequence, while a dashed arc shows that the note interval is expanded by the fuzzy rules. We use all the connected n-gram paths in the graph to query the index of the melody database.

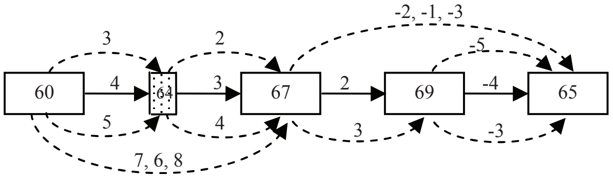

Figure 2. An example of the expanded graph of note intervals.

## 2.3. Candidate pruning

After the transcription error compensation, the query n-grams are used to query the index to get a list of match candidates. Since only the tone feature is used in n-gram extraction, the candidates need to be pruned based on the rhythm distance. For each candidate, suppose the duration vector of the associated n-gram notes of the query is $[d(q_1), d(q_2), \cdots, d(q_n)]$, and the duration vector of the n-gram notes of the candidate melody is $[d(c_1), d(c_2), \cdots, d(c_n)]$, then the rhythm distance is calculated by the following function:

$$dist(d(q),d(c)) = \sum_{i=1}^{n}(d^{norm}(q_i) - d^{norm}(c_i))^2 \quad (1)$$

where $d^{norm}(q_i) = \frac{d(q_i)}{\sum d(q_i)}$ and $d^{norm}(c_i) = \frac{d(c_i)}{\sum d(c_i)}$ ( $0 < i \le n$ ). Candidates with rhythm distance larger than a pre-defined threshold are discarded.

## 3. QBH PROTOTYPE SYSTEM

A framework of the QBH system is shown in Figure 3. Generally, the QBH system works like this: first, a query is input to the system via a microphone, and then transcribed into a sequence of notes. Second, the n-gram fast match is performed to generate a list of match candidates. Finally, the accurate match is performed on the pitch contour, and rescores the candidates to give the top-N list.

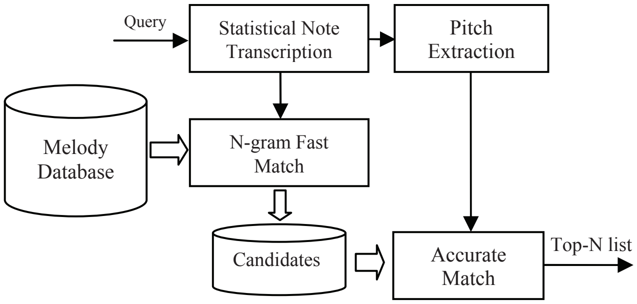

Figure 3. Paradigm of the prototype system.

### 3.1. Statistical note transcription

In our QBH system, we used the statistical note transcription algorithm [7]. In order to interpreting the note sequence as a word sequence, the algorithm just trained a “customized” acoustic model and language model, and then plugged the models into a speech recognition engine to decode the signal into a sequence of notes. The acoustic model was trained with higher-order cepstral features instead of f0 to avoid the hard-decision that has to be made in the explicit voiced-unvoiced segmentation and pitch extraction, and a key-independent 4-gram model was trained as the language model. Our previous experiments had showed that the statistical algorithm outperformed three other state-of-the-art note transcription systems in various acoustic conditions.

### 3.2. Pitch extraction

A modified autocorrelation algorithm is used to extract the pitch contour from the query. After we got the note transcription result, for each frame of the signal, a reliable voiced/unvoiced classification can be obtained. We can also know the tone if the frame is voiced. Thus, the pitch extraction is only performed for a voiced frame, and it searches for the pitch related peak only in the area associated with the tone label rather than search globally. Thus, the pitch extraction can heritage the advantage of the note transcription algorithm and tends to be robust against noise.

### 3.3. Accurate match algorithm

In the QBH system, before the accurate match is performed, the candidate melodies need to be normalized to eliminate the difference of key scale and tempo. Given a candidate returned by the fast match, the key difference and tempo ratio can be easily estimated by comparing the absolute tone and duration of the n-gram notes of the query with the candidate melody.

The accurate match is a phrase-level recursive alignment (PHRA) algorithm, motivated by the fact that singers usually keep coherent tempo within a musical phrase, but are more likely to breathe at phrase boundaries and thus deviate from the perfect rhythm. The algorithm first segments the query into several fragments, each of which roughly corresponds to a musical phrase. Then, it assumes that the alignment path within each phrase is a straight line, and allows for rhythm variations only at the phrase boundaries.

Similar with the recursive alignment (RA) algorithm proposed in [4], the PHRA algorithm also searches for the optimal alignment path in a top-down fashion. The query contour is first matched with the candidate melody in a global view before split into two parts to have more local match. The split procedure is recursively repeated until the maximal recursion depth is reached. The PHRA algorithm differs with the original RA algorithm in splitting the query contour only at an estimated phrase boundary. Experiments showed that the PHRA algorithm outperformed other existing accurate match algorithms like the linear scaling (LS), dynamic time warping (DTW), and RA.

## 4. EXPERIMENTAL RESULTS

### 4.1. Database

We evaluated our approach on the ThinkIT QBH query corpus. It is one of the evaluation sets used in 2008 MIREX QBH competition[1]. The query set contains 355 queries recorded by ordinary singers, most of which are sung with lyrics, and covers 106 melodies of Chinese popular songs. The average length of the queries is 12.4 s. All data are sampled at 8 kHz at 16 bit rate PCM.

The melody database contains two parts of data. The first part is the 106 target songs of the query data, given in MIDI file from ThinkIT. The second part is 5,117 EsAC[2] melodies as noises. In total the database contains 5,223 melodies, and we segmented the melodies into 31,412 musical “sentences” or “phrases”. In general, the humming of human usually starts from natural boundary of a certain sentence of lyric and stops in the middle of or at the end of a sentence. Therefore, our system assumes that the query starts from the beginning of a melodic segment, but could end at any point in the melody.

### 4.2. Evaluation results

In our experiments, we measured the accuracy of melody match by top-1 rate and mean reciprocal rate (MRR). MRR is calculated as the average reciprocal rank of the target melody. The MIREX competition uses MRR as a standard measurement of QBH performance in for several years:

[1] About MIREX QBH evaluation, please see: http://www.music-ir.org/mirex/2008/index.php/Main_Page

[2] EsAC database is available from: http://www.cs.uu.nl/events/dech1999/dahlig/

$$MRR = \sum_{q=1}^{N} \frac{1}{c_q} \qquad (2)$$

where $c_q$ is the rank of the target melody for the q-th query. We counted those whose target melody within the top-20 returns. If the target melody was out of the top-20 returns, the corresponding $1/c_q$ was set to 0. The maximal value of MRR is 1.0 and can be achieved if all targets are returned as the top-1.

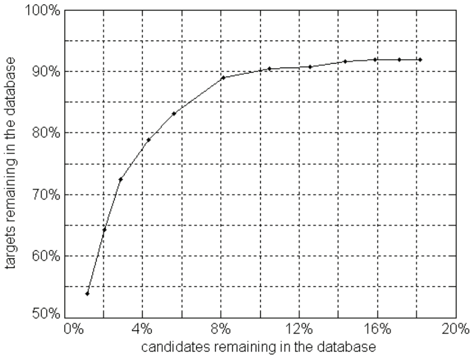

Figure 4. The performance of n-gram as fast match method of the QBH system

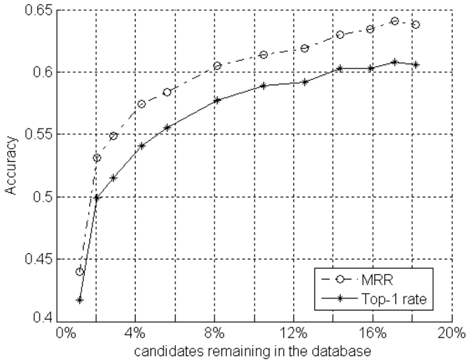

Figure 5. The accuracy of the QBH system when the n-gram fast match was used.

Through adjusting the number of n-grams extracted from the query and the distance thresholds used in the candidate pruning of the fast match, we were able to tune the percentage of the candidates remaining in the database for accurate match. Hence we could see whether it can hold the target melodies even if very few candidates returned in fast match stage.

Figure 4 illustrates the performance of n-gram fast match. It is clear that when 10% of the candidates remain in the database after fast match stage, which means the computation amount of accurate match is reduced by 10 times, about 90% of the target melodies are still preserved in the candidates. Even when only 5% of the database remains, the percentage of the targets preserved is still close to 80%.

When no fast match used and the accurate match is performed directly on the whole database, the top-1 rate was 0.67 and the MRR was 0.70. Figure 5 shows the accuracy of the system when the fast match was used. If 10% of the database is returned by the fast match, the top-1 rate was 0.59 and the MRR was 0.61. The relative accuracy loss was about 12%. If 5% of the database is returned, the top-1 rate was 0.54 and the MRR was 0.57. The relative accuracy loss was about 19%. The system performance data is consistent with the fast match performance data shown in figure 4.

## 5. CONCLUSIONS

This paper presented an approach of effective n-gram fast match for Query-by-Humming system. Compared with other relevant studies, the advantage of our n-gram method comes from both the use of a robust statistical note transcription algorithm and a novel error compensation method. Our approach uses fuzzy rules to deal with note recognition errors, so as to save the total number of computation of accurate matches while holding a high recalling rate. The experiments on a relatively large database showed that the n-gram fast match had a satisfactory performance. When the database was reduced to 10% of the whole size, about 90% of the target melodies were still preserved. Meanwhile, about 88% of the match accuracy of system was kept.

The performance of the n-gram match could be further improved by compensating the note transcription errors more accurately with data-driven rules instead of the current manual rules. It is also important to develop other match algorithms and use them in conjunction with the n-gram algorithm to optimize the system performance.

## 6. ACKNOWLEDGEMENTS

We would like to thank Liu Wen of IBM China Research Laboratory, Beijing, for his help in running experiments. This work was supported by joint research grant of IBM and Tsinghua University 2007-2008.

## 7. REFERENCES

[1] R.B. Dannenberg, W.P. Birmingham, G. Tzanetakis, et al., “The MUSART Testbed for Query-by-Humming Evaluation,” Proc. of International Symposium of Music Information Retrieval (ISMIR), 2003.

[2] J.R. Jang and M.Y. Gao, “A Query-by-Singing System based on Dynamic Programming,” Proc. of International Workshop on Intelligent Systems Resolutions, pp. 85-89, Dec. 2000.

[3] J. Shifrin and W. Birmingham, “Effectiveness of HMM-based Retrieval on Large Databases,” Proc. of International Symposium of Music Information Retrieval (ISMIR), 2003.

[4] X. Wu, M. Li, J. Liu, et al. “A Top-down Approach to Melody Match in Pitch Contour for Query by Humming,” Proc. of International Symposium on Chinese Spoken Language Processing, Dec. 2006.

[5] A.L. Uitdenbogerd and J. Zobel, “Music Ranking Techniques Evaluated,” Proc. of the 25th Australasian Computer Science Conference (ACSC2002), 2002.

[6] R.B. Dannenberg and N. Hu, “Understanding Search Performance in Query-by-Humming Systems,” Proc. of International Symposium of Music Information Retrieval (ISMIR), 2004.

[7] D.N. Jiang, M. Picheny, and Y. Qin, “Voice-melody Transcription under a Speech Recognition Framework,” Proc. of ICASSP 2007.