Query by singing and humming system based on combined DTW and Linear Scaling

Hsin-Cheng Lee, Pao-Chi Chang
Dept. Communication Engineering, National Central University, Taiwan

***Abstract*—In this paper, a query by singing and humming system based on combined DTW and Linear Scaling is proposed. Users can search for songs in the database by singing or humming some parts of a songs. Firstly, Yin method [9] is used to estimate the query's fundamental frequence, then LS is employed to scale down the pitch sequence to reduce the computation time. In the matching stage, Dynamic Time Warping (DTW) is applied to find the Top-10 most matching songs from the database.**

I. Introduction

Query By Singing and Humming (QBSH) system is a kind of content-based music information retrieval system. People can sing or hum some parts of the song that we want to search, and the QBSH system will return the Top-k matched songs from the database. There are many previous works on QBSH, Nam et al. [1]-[4] discussed the combination of different classifiers and the score fusion methods, however, a poor classifier often reduces the overall accuracy. Hu et al. [5] used Hidden Markov Model and add the tempo features for recognition. Liu et al. [6] used auto-encoder as a feature extractor combined with local sensitive hashing method to reduce the computation time. Stasiak. [7] used adaptive tuner to improve user's bad singing skills, but it also increased the error rate. Sun. et al. [8] used DBN as a matcher to retrieve songs in database.

II. Proposed Method

In this paper, the proposed system is shown in Fig. 1, our system mainly comprised of three parts, preprocessing module, matching module and refinement module.

The preprocessing module consists of pitch tracker and smoothing method. In pitch tracker, Yin method [9] is used to estimate the fundamental frequence of the query and we also convert the fundamental frequence into MIDI numbers or called "pitch". In the smoothing method, when the pitch is unreasonable, we mark it as "error fragment" or "large jump" and set them to be the same values as the last non-zero pitch, we also use median filter of the order of 3 frames to smooth the pitch sequence, as shown in Fig. 2.

The matching module is shown in the blue block in Fig. 1. In order to reduce the computation time, before entering the matching module, the system will scale down the query's pitch sequence by linear scaling, then match the query's singing or humming beginning position corresponding to the reference MIDI data by sliding the matching window by 30 frame each time in MIDI, we calculate the DTW values between the query and all the sliding matching windows, the beginning position is the index where the minimum value of DTW appears, as shown in Figure 3. After finding the beginning position, we also shrink or stretch multiple matching windows determined by DTW to get a more accurate result , and obtain a DTW_{minj}, this value represents the final similarity between query and the j-th MIDI in database.

The refinement module is shown in the red block in Fig. 1. When DTW_{minj} is less than threshold, enter the refinement module, we use LS to scale down the MIDI and recalculate a $\mathrm{DTW}_{refine-j}$ by the same steps with DTW_{minj} , the new similarity between query and the j-th MIDI is the smaller value between DTW_{minj} and $\mathrm{DTW}_{refine-j}$.The relationship between matching module and refinement module is shown in Figure 4.

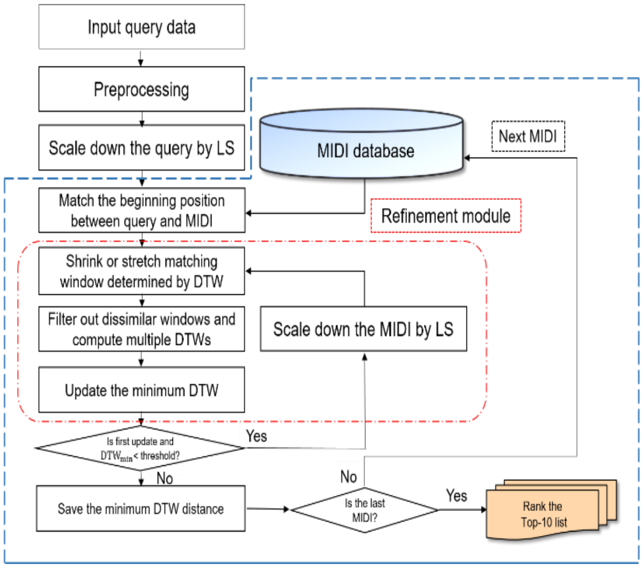
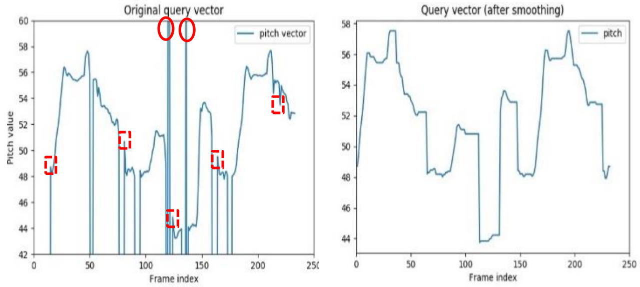

Input query data
Preprocessing
Scale down the query by LS
MIDI database
Next MIDI
Match the beginning position between query and MIDI
Refinement module
Shrink or stretch matching window determined by DTW
Filter out dissimilar windows and compute multiple DTWs
Scale down the MIDI by LS
Update the minimum DTW
Is first update and DTW$_{min}$< threshold?
Yes
No
Save the minimum DTW distance
No
Is the last MIDI?
Yes
Rank the Top-10 list

Figure 1. The proposed QBSH system.



(a) (b)

Figure 2. The result of smoothing method: (a) the original pitch vector, (b) the pitch vector after smoothing.

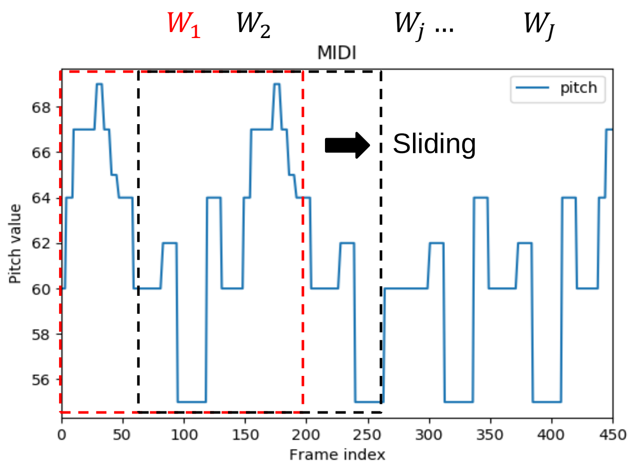


Figure 3. Match the beginning position by sliding the matching window in MIDI.

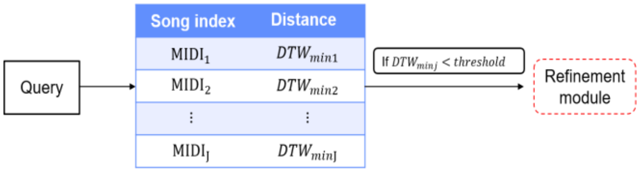
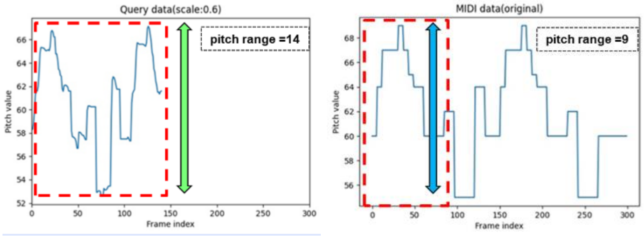


Figure 4. The relationship between refinement module and matching module.



(a) (b)

Figure 5. (a) the pitch range of query, (b) the pitch range of the j-th matching window in MIDI.

We also design a pre-filter to filter out the dissimilar matching window as shown in Fig. 5, when the pitch range difference between query and the j-th matching window W_j larger than the threshold, we would filter out the window.

III. EXPERIMENTAL RESULTS

In our experiment, we use two datasets. The first database (denoted by dataset1) is the publicly available MIR-QBSH corpus [10] which consists of 48 ground-truth MIDI files and 4431 singing/humming queries as wav files with the sampling rate of 8000 Hz. The next one, denoted by dataset2, is dataset 1 mixed with 2000 ground-truth MIDI files from Esssen collection [11].

The evaluation measurements are Top-K accuracy, mean reciprocal rank (MRR), and the computation time (each query). The Top-K accuracy is the percentage that the correct song is ranked within the Top K songs list returned by system.

We test the dataset1 and compare our performance with other methods, as shown in Table 1.

We also do the retrieval experiment on dataset2, the results are shown in Table 2.

IV. CONCLUSION

In this paper, a QBSH system combined DTW and LS is proposed. Our experimental result on MIR-QBSH corpus reaches the MRR 0.958 which is much higher than other researches. Our system can also maintain good performance on dataset2 which is much larger than dataset1.

Table 1
Performance comparison of the proposed method with other researches on the dataset1.

Method	Top1(%)	Top10(%)	MRR(%)	Time(s)
Baseline [1]	77.27	85.56	79.3	-
DTW+QLS [2]	70.14	86.16	74.6	-
QDTW+LS [3]	79.14	89.77	81.9	0.04
Multi-classifier [4]	84.15	87.58	85	-
HMM+TDP [5]	74.15	94.38	80.6	-
Proposed method				
Without refinement module	91.2	96.86	93.13	0.07
Use refinement module	94.43	98.22	95.81	0.125

Table 2
The experimental result on the dataset2

Dataset	Top1(%)	Top3(%)	Top10(%)	MRR(%)
Dataset2	87.09	90.68	93.57	89.28

REFERENCE

[1] G. P. Nam, K. R. Park, S.-J. Park, T. T. T. Luong, and H. H. Nam, “Intelligent query by humming system based on score level fusion of multiple classifiers”, EURASIP 2011

[2] G. P. Nam, K. R. Park, S.-J. Park, S.-P. Lee, and M. Y. Kim, “A new query by humming system based on the score level fusion of two classifiers”, Int J Comm Syst 2012

[3] G. P. Nam and K. R. Park, "Fast query-by-singing/humming system that combines linear scaling and quantized dynamic time warping algorithm“, IJDSN 2015

[4] G. P. Nam and K. R. Park, "Multi-Classifier Based on a Query-by-Singing/Humming System", Computer Science Symmetry 2015

[5] C. W. Lin, J. J. Ding, and C. M. Hu, “Advanced query by humming system using diffused hidden Markov model and tempo based dynamic programming”, APSIPA ASC 2016

[6] A. Lv and G. Liu, “AnEffective Design for Fast Query-by-Humming System with Melody Segmentation and Feature Extraction”, ICCSEC 2017

[7] B. Stasiak, “Follow That Tune – Dynamic Time Warping Refinement for Query by Humming”, Proc. Of Joint Conference NTAV/SPA 2012.

[8] J. Q. Sun andS. P. Lee, “Query by singing/humming system based on deep learning”, IJAER 2017

[9] A. De Cheveigné and H. Kawahara, “YIN, a fundamental frequency estimator for speech and music”, J. Acoust. Soc. Amer 2002

[10] MIR-QBSH corpus: http://www.music-r.org/mirex/wiki/2009:Query_by_Singing/Humming

[11] Essen collection: http://www.esac-data.org