Report
Query by Humming
Meghana Sudhindra

Query by humming (**QbH**) is a music retrieval system that branches off the original classification systems of title, artist, composer, and genre. It normally applies to songs or other music with a distinct single theme or melody. The system involves taking a user-hummed melody (input query) and comparing it to an existing database. The system then returns a ranked list of music closest to the input query [1].

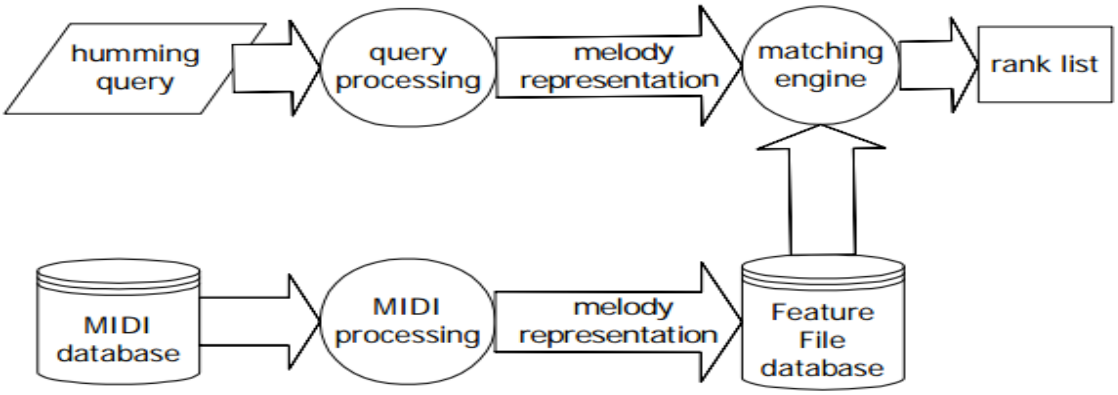

Figure – 1: Query by humming flow chart

State of the Art

Most of recent works on QbH is focused on melody representations, similarity measures and matching processing. In some works, such as [1][2], only pitch contour is used to represent melody. Typically, QbH approaches firstly extract the F0 contour and/or a note-level transcription for a given vocal query, and then a set of candidate melodies are retrieved from a large database using a melodic matcher module [3]. In McNab *et al* [4], pitch interval and rhythm are considered as well as pitch contour. In [5], relative interval slope is used in music information retrieval. And [6] introduces four basic segment types (A, B, C, D) to model music contour. The challenges related to QBSH are [7]: reliable pitch tracking in noisy environments, automatic song database preparation (predominant melody extraction and transcription), efficient search in very large music collections, dealing with errors of intonation and rhythm in amateur singers, etc. In another method, a new distance metrics [8] between query and songs is proposed. But its computation is very time-consuming because it need adjust many parameters step by step to find the minimum distance. In fact, not only should melody representation maintain melody information as much as possible, but also it should suit the habits of people humming.

In this report, we analyse the performance of various state-of-the-art F0 trackers for QBSH in different conditions of background noise and smartphone-style distortion.

For this study, we have considered three different melodic matchers: two state-of-the-art systems (one of which obtained the best results in MIREX 2013), and a simple, easily reproducible baseline method based on frame-to-frame matching using dynamic time warping (DTW).

Datasets

The datasets for Query by Humming are plenty but open, annotated and structured datasets are limited. But I found two most widely used datasets which are used 64 times in MIREX from 2006 to 2015. The main drawback of both datasets is its both majorly Chinese singing datasets but it's well annotated.

IOACAS QBH dataset

This dataset has 298 ground truth and 759 singing/humming data. This dataset is created by Institute of Acoustics, Chinese Academy of Sciences. The ground truth has piano recordings mostly of Chinese pop songs and the query generally by a female voice. This has been widely used in MIREX submissions from 2006 to 2015.

MIR-QBSH-corpus

This dataset has 48 Ground truth and 4431 singing/humming data created by MIR (Multimedia Information Retrieval) Lab at CS Dept. of NTHU (National Tsing Hua University), Taiwan. The ground Truth has piano recordings of both English Rhymes and Chinese songs and the query of both female and male voice. The dataset also contains an additional pitch vector file containing manually labelled pitch file, with frame size = 256 and overlap = 0. The link to. These datasets are totally used 64 times from 2006 to 2015. The statistics are shown below:

https://docs.google.com/spreadsheets/d/1eawIM0NPqNM08HgSsOe2Bn56Ht4VLIN1aFaXOlnTSws/edit#gid=0

The other datasets which was stumbled upon on the task were MTG QBH dataset (no MIDI file), TANSEN (not open dataset) and Tunebot Dataset (not open dataset).

F0 Trackers

The fundamental frequency (F0) of a periodic signal is the inverse of its period, which may be defined as the smallest positive member of the infinite set of time shifts that leave the signal invariant. This definition applies strictly only to a perfectly periodic signal, an uninteresting object (supposing one exists!) because it cannot be switched on or off or modulated in any way without losing its perfect periodicity. Interesting signals such as speech or music depart from periodicity in several ways, and the art of fundamental frequency estimation is to deal with them in a useful and consistent way

In this section the we look at the different approaches of tracking F0 and consequently matching with the given queries.

YIN Algorithm was developed by de Cheveign´e and Kawahara in 2002 [9]. This introduces a method for F0 estimation that produces fewer errors than other well-known methods. The name YIN (from ‘‘yin’’ and ‘‘yang’’ of oriental philosophy) alludes to the interplay between autocorrelation and cancellation that it involves.
pYIN Algorithm method has been published by Mauch in 2014 [10], and it basically adds a HMM-based F0 tracking stage to find a “smooth” path through the fundamental frequency candidates obtained by Yin.
Swipe Algorithm was published by A. Camacho in 2007 [11]. This algorithm estimates the pitch as the fundamental frequency of the saw tooth waveform whose spectrum best matches the spectrum of the input signal. The algorithm proved to outperform other well-known F0 estimation algorithms, and it is used in the F0 estimation stage of some state-of-the-art query-by-humming systems.
MELODIA is a system for automatic melody extraction in polyphonic music signals developed by Salamon and Gomez in 2012 [12]. This system is based on the creation and characterisation of pitch contours, which are time continuous sequences of pitch candidates grouped using auditory streaming cues. Melodic and non-melodic contours are distinguished depending on the distributions of its characteristics.

Approaches
Audio to MIDI melody matchers
In this section the description of the Baseline Method used for audio-to-MIDI melodic matching [13] is presented. This method is implemented with a simple, freely available baseline approach based on dynamic time warping (DTW) for melodic matching and consists of four steps (a scheme is shown in Figure 2):

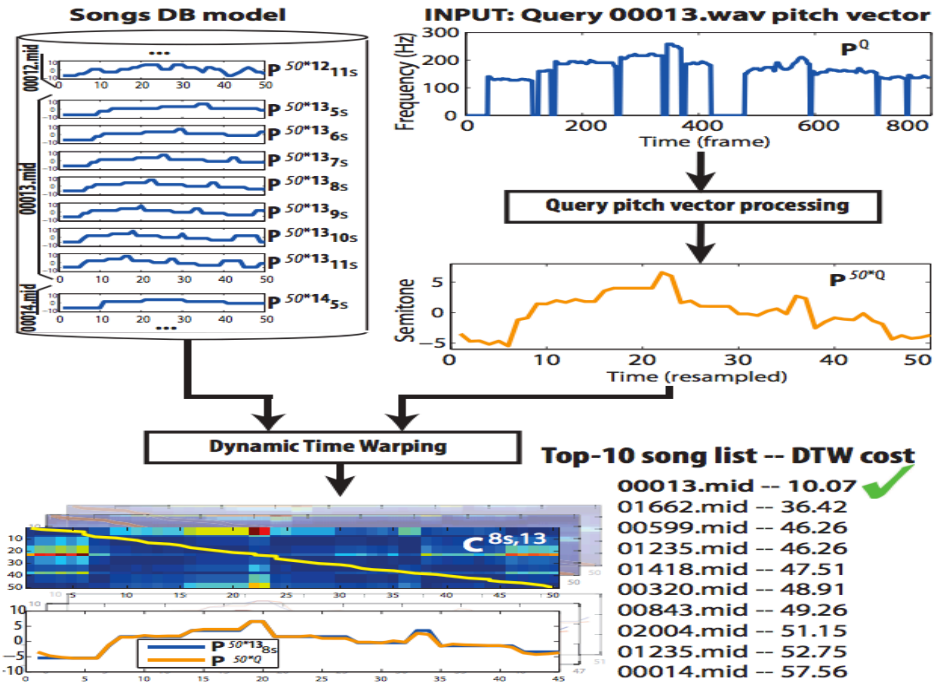

Figure 2. Scheme of the proposed baseline method for audio-to-MIDI melody matching [13].
The four steps followed are [13]:

(1) **Model building**: We extract one pitch vector (in MIDI number) for every target MIDI song using a hop-size of 0.01 seconds. Then we replace unvoiced frames (rests) in pitch vector by the pitch value of the previous note, except for the case of initial unvoiced frames, which are directly removed. Then, each pitch vector is truncated to generate 7 pitch vectors with lengths [500, 600, 700, 800, 900, 1000, 1100] frames (corresponding to the first 5, 6, 7, 8, 9, 10 and 11 seconds of the target MIDI song, which are reasonable durations for a user query. Finally, all these pitch vectors are resampled (through **linear interpolation**) to a length of 50 points, and then zero-mean *normalized* (for a *common key transposition*). The vector obtained will be P^{50*k}

(2) **Query pre-processing**: The pitch vector of a given .wav query is loaded (note that all pitch vectors are computed with a hopsize equal to 0.01 seconds). Then, unvoiced frames are replaced by the pitch value of the previous note, except for the case of initial unvoiced frames, which are directly removed. This processed vector is then converted to MIDI numbers with 1 cent resolution, and labelled and is resampled (using linear interpolation) to a length L = 50 and zero-mean normalized (for a common key transposition). The vector obtained will be P^{50*q}

(3) **DTW-based alignment**: Now we find the optimal alignment between P^{50*q} and all pitch vectors P^{50*k} using dynamic time warping (DTW). In our case, each cost matrix $C^{Duration,k}$is built using the squared difference:

$$C^{Duration,k}(i,j) = (P^{50*q}(i) - P^{50*k}(i))^2$$

Where k is the target song index, Duration represents the truncation level (from 5s to 11s), and i, j are the time indices of the query pitch vector P^{50*q} and the target pitch vector P^{50*k} duration respectively. The optimal path is now found using Dan Ellis' Matlab implementation for DTW [14] with the following allowed steps and associated cost weights [Δi, Δj, W]: [1, 1, 1], [1, 0, 30], [0, 1, 30], [1, 2, 5], [2, 1, 5]. The allowed steps and weights have been selected to penalize 0 or 90 angles in the optimal path (associated to unnatural alignments), and although they lead to acceptable results, they may not be optimal.

(4) **Top-10 report**: Once the P^{50*q} has been aligned with all target pitch vectors (a total of 7 × N songs vectors, since we use 7 different durations), the matched pitch vectors are sorted according to their alignment total cost. Finally, the 10 songs with minimum cost are reported.

Music Radar's approach

MusicRadar [15] is a state-of-the-art algorithm for melodic matching, which participated in MIREX 2013. The system takes advantage of several matching methods to improve its accuracy. First, Earth Mover's Distance (EMD), which is note-based and fast, is adopted to eliminate most unlikely candidates. Then, Dynamic Time Warping (DTW), which is frame-based and more accurate, is executed on these surviving candidates. Finally, a weighted voting fusion strategy is employed to find the optimal match. In my study, I

have used the exact melody matcher tested in MIREX 2013, provided by its original author.

Evaluation

Datasets

In addition to MIR QBSH and IOACAS, I have used the Audio Degradation Toolbox [16] in order to recreate common environments where a QBSH system could work. Specifically, we have combined three levels of pub-style added background noise (PubEnvironment1 sound) and smartphone-style distortion (smartphone Recording degradation), leading to a total of seven evaluation datasets: (1) Original MIRQBSH corpus (2) 25 dB SNR (3) 25 dB SNR + smartphone distortion (4) 15 dB SNR (5) 15 dB SNR + smartphone distortion (6) 5 dB SNR (7) 5 dB SNR + smartphone distortion. Note that all these degradations have been checked in order to ensure perceptually realistic environments.

Before commenting on the evaluation, the dataset IOACAS and MIR have different way a song is queried. Every data set has a midi folder (original song) and query(.wav) from the user. Generally, the queries are in the format of ta da ta ra (easier for the evaluation and F0). Just for the user experience I have provided links to one of the MIDI and QUERY to understand it better. Also after performing this algorithms/approaches I found out that my approaches were suitable for only MIR QBSH dataset than IOACAS dataset. This was firmly confirmed after listening to the queries that in the former the queries are in the beginning and in the latter queries are anywhere in the file hence the process is completely different. Also, IOACAS dataset didn't have a pitch vector file so a python script had to be written and to output the pitch vector in the required format to perform the task in MATLAB. Hence the results of MIR QBSH will be shown significantly.

Evaluation Measures

(1) **Mean overall accuracy of F0 tracking** $(\boldsymbol{ACC_{ov}})$ For each pitch vector we have computed an evaluation measures defined in MIREX Audio Melody Extraction task: The mean overall accuracy is then defined as $ACC_{ov} = (\frac{1}{N}) \sum_{i=1}^{N} ACC_{ovi}$ where N is the total number of queries considered and ACC_{ovi} is the overall accuracy of the pitch vector of the i:th query. We have selected this measure because it considers both voicing and pitch, which are important aspects in QBSH. For this measure, our ground truth consists of the manually corrected pitch vectors of the .wav queries, which are included in the original MIR QBSH corpus.

(2**) Mean Reciprocal Rank (MRR**): This measure is commonly used in MIREX Query By Humming task and it is defined as: $MRR = (\frac{1}{N}) \sum_{i=1}^{N} {r_i}^{-1}$ where N is the total number of queries considered and $r_i$is the rank of the correct answer in the retrieved melodies for *i:*th query.

F0 Trackers	Clean Dataset	25dB SNR	25dB SNR + Dist	15dB SNR	15dB SNR + Dist	5dB SNR	5dB SNR + Dist
Manually corrected	100/0.82/0.96	100/0.82/0.96	100/0.82/0.95	100/0.82/0.96	100/0.82/0.96	100/0.82/0.96	100/0.82/0.95
YIN	86/0.62/0.89	86/0.7/0.92	81/0.64/0.89	82/0.62/0.89	75/0.5/0.82	48/0.03/0.04	44/0.04/0.03
pYIN	90/ 0.71/0.92	90/ 0.74/ 0.93	85/ 0.74/0.94	90/0.78/ 0.94	85/0.77/ 0.94	79/0.69/ 0.87	72/0.58/0.81
SWIPE	89/0.71/0.92	89/0.71/0.92	84/0.66/0.91	88/0.72/0.93	83/0.65/0.91	75/0.67/0.82	66/0.48/0.73
MELODIA MONO	87/0.66/0.87	87/0.67/0.87	83/0.64/0.84	86/0.66/0.84	82/0.58/0.80	83/0.51/0.75	73/0.32/0.62

Table 1: F0 overall accuracy and MRR obtained for each case. The format of each cell is: $\boldsymbol{ACC_{ov}}$(%) / MRR-baseline / MRR-MusicRadar.

Results and Conclusions

In Table 1, we show the $\boldsymbol{ACC_{ov}}$ and the MRR obtained for the whole dataset of 4431.wav queries in each combination of F0 tracker-dataset-matcher (189 combinations in total). As expected, the manually corrected pitch vectors produce the best MRR in most cases (the overall accuracy is 100% because it has been taken as the ground truth for such measure). Note that, despite manual annotations are the same in all datasets, MusicRadar matchers do not produce the exact same results in all cases. It is due to the generation of the indexing model (used to reduce the time search), which is not a totally deterministic process.
Additionally, we observed that, in most cases, the queries are matched either with rank 1 or rank ≥ 11 (intermediate cases such as rank = 2 or 3 are less frequent). Therefore, the variance of ranks is generally high, their distribution is not Gaussian.

Robustness
In order to study the robustness of each melodic matcher to F0 tracking errors, we take queries of which produce the right answer in first rank for the two matchers considered (baseline, Music Radar) when manually corrected pitch vectors are used (around a 70% of the dataset matches this condition). In this way, we ensure that bad singing or a wrong manual annotation is not affecting the variations of MRR. Given that the baseline matcher only uses DTW, whereas the MRR matcher use a combination of various searching methods we hypothesise that such combination may improve their robustness to F0 tracking errors. Further research may be done to investigate this.

By performing this experiment, I found out that pYIN algorithm outperforms all of them due to its approach and the HMM based FO tracking. Of the two approaches used Music Radar is better than the Baseline Approach due to taking both DTW and EMD in account.

Acknowledgements

I would like to thank Emilia Gomez from UPF, Emilio Molina from BMAT, Rong Gong and Siddharth Bhardwaj from UPF for timely help, corrections and contributions.

References

[1] A.Ghias, J.Logan, D.Chamberlin & B.C.Smith "Query by humming-musical information retrieval in an audio database", ACM Multimedia,1995

[2] S. Blackburn and D. DeRoure, "A Tool for Content Based Navigation of Music", Proc. ACM Multimedia98, pp 361- 368, 1998.

[3] L. Wang, S. Huang, S. Hu, J. Liang and B. Xu: "Improving searching speed and accuracy of query by humming system based on three methods: feature fusion, set reduction and multiple similarity measurement rescoring," Proceedings of INTERSPEECH, 2008.

[4] R. J. McNab, et al, "Towards the Digital Music Library: Tune Retrieval from Acoustic Input". Proc. of Digital Libraries, pp 11-18, 1996.

[5] K. Lemstrom, P. Laine and S. Perttu. "Using Relative Interval Slope in Music Information. Retrieval". In Proc. of International Computer Music Conference 1999(ICMC '99), pp. 317-320, 1999.

[6] A. L.P. Chen, M. Chang and J. Chen. "Query by Music Segments: An Efficient Approach for Song Retrieval". In Proc. of IEEE International Conference on Multimedia and Expo., 2000.

[7] J. -S. Roger Jang: "QBSH and AFP as Two Successful Paradigms of Music Information Retrieval" Course in *RuSSIR*, 2013.

[8] C. Francu and C. G. Nevill-Manning. "Distance Metrics and Indexing Strategies for a Digital Library of Popular Music". In Proc. of IEEE International Conference on Multimedia and Expo. 2000.

[9] A. De Cheveign´e and H. Kawahara: "YIN, a fundamental frequency estimator for speech and music," Journal of the Acoustic Society of America, Vol. 111, No. 4, pp. 1917-1930, 2002.

[10] M. Mauch, and S. Dixon, "pYIN: A fundamental frequency estimator using probabilistic threshold distributions," Proceedings of ICASSP, 2014.

[11] A. Camacho: “SWIPE: A sawtooth waveform inspired pitch estimator for speech and music,” PhD dissertation, University of Florida, 2007.

[12] J. Salamon and E. G ´omez: “Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics,” IEEE Transactions on Audio, Speech and Language Processing, Vol. 20, No. 6, pp. 1759–1770, 2012.

[13] Molina, Emilio and Tardon, Lorenzo J and Barbancho, Isabel and Barbancho “The importance of F0 tracking in query-by-singing-humming,” Proceedings of the International Symposium on Music Information Retrieval, pp.227–282, 2014.

[14] D. Ellis: “Dynamic Time Warp (DTW) in Matlab”, 2003. Web resource, available: www.ee.columbia.edu/ dpwe/resources/matlab/dtw/

[15] Doreso Team (www.doreso.com): “MIREX 2013 QBSH Task: Music Radar’s Solution” Extended abstract for MIREX, 2013.

[16] M. Mauch and S. Ewert: “The Audio Degradation Toolbox and its Application to Robustness Evaluation,” Proceedings of ISMIR, 2013.