Improving Query by Humming System using Frequency-Temporal Attention Network and Partial Query Matching

Muhammad Ulfi
School of Electrical Engineering and Informatics
U-CoE AI-VLB
Institut Teknologi Bandung
Bandung, Indonesia
23520001@mahasiswa.itb.ac.id

Rila Mandala
School of Electrical Engineering and Informatics
U-CoE AI-VLB
Institut Teknologi Bandung
Bandung, Indonesia
rila@itb.ac.id

Abstract*— Search engine technology makes it easy to find information from the many available sources. One of the available information is music. In Music Information Retrieval, Query by Humming is an effective and natural method for searching music in databases. The Unified Algorithm is the latest research in this field. The QbH system is divided into two stages: melody extraction and matching. Several studies have found that the statistical approach used in the Unified Algorithm did not perform better than the data-driven approach. In addition, the song that the user hums often is only a part of the whole melody. This causes a match to be made between the humming query and the query containing the entire melody. This study proposes a modification to the QbH system by using a data-based approach for melody extraction and matching only a part of the melody. The latest research on the topic of data-based melody extraction is the Frequency-Temporal Attention Network (FTANet). The combination of FTANet as melody extraction and modification of the unified algorithm can provide better performance compared to the baseline system. However, in terms of computational time, both melody extraction and melody matching processes take much longer. This study also revealed that the Unified algorithm is not suitable for use in systems with large datasets.

Keywords—music information retrieval, query by humming, melody extraction, melody matching, melody similarity

I. INTRODUCTION

As time goes by, technological developments will become more advanced and the amount of information available will increase. At this time, most people use search engine technology to find the information they want, mainly because of the ease of access to the internet network and providing fast search results. Various kinds of information are available on the internet network, one of which is music. Query by Humming (QbH) is an effective and natural method of searching for music in databases [1]. Music search is done by matching the results of humming with music in the database. The QbH system uses the user's hum or song as an input to the system to find the desired music. The tone from user input will be converted into a series of transitions from the pitch as a query, then search for music in the database using a query engine, and the system will provide a list of music ratings from the search results.

Recent research on this topic is Unified Algorithm [2]. An autocorrelation-based method [3] is used for melody extraction in this research. The Unified Algorithm is a combination of 3 n-gram algorithms and Mode Normalized Frequency (MNF), which is a string representation of the melody pitch with the letter N as a reference. As the song that the user hums is often only part of the whole melody, this algorithm makes a match between the humming query and the query containing the entire melody.

This study resulted in a mean reciprocal rank (MRR) of 0.59. However, there is still room for improvement, such as using other methods for melody extraction and making changes to the matching method to give better results.

This research purposes frequency-temporal attention network (FTANet) [4], which is a data-based approach as a melody extraction method. This research shows a better overall performance in general compared to existing state-of-the-art methods for melody extraction. Then, changes were made to the unified algorithm to match the subquery section to provide better results. The combination of FTANet as melody extraction and the modified unified algorithm is expected to perform better than in the previous study.

II. RELATED WORK

The QbH system is divided into two phases, namely melody feature extraction and melody matching [1]. In other studies, there are additional steps for data cleaning of user hums, such as eliminating noise and segmenting melodies before feature extraction is performed [2].

Several QbH systems have been studied previously, including a QbH system using a combination of Harmonic Enhancement and Note Segmentation for melody extraction, and Dynamic Programming Matching as a melody matching technique [5], a combination of Hidden Markov Model (HMM) and Convolutional Neural Network (CNN) as melody extraction, and the matching technique using Local Sensitive Hashing (LSH), Earth Mover's Distance (EMD), and Dynamic Time Warping. (DTW) [6], and a combination of melody extraction technique using an autocorrelation-based method [3], and matching technique using the Unified Algorithm, which consists of four algorithms, namely Relative Pitch 4-Grams (RP4G), 3-Grams (RP3G), and 2-Grams (RP2G), and Mode Normalized Frequency (MNF) [2].

A study [7] conducted a comparison between several algorithms for extracting melodies. The evaluation is measured using four error matrices, namely Gross Pitch Error (GPE), Fine Pitch Error (FPE), Voicing Decision Error (VDE), and F0 Frame Error (FFE). The autocorrelation-based method used in recent research [2] only has the best value on VDE, which shows the proportion of errors in determining

978-1-6654-5668-5/22/$31.00 ©2022 IEEE

the presence or absence of sound in the frame, while on the other three matrices, other algorithms show better performance.

Research [8] stated that the features of a simple and unoptimized neural network architecture could outperform hand-crafted features built on expert knowledge when comparing approaches using hand-crafted and data-based approaches using CNN to classify the basic frequency contours. Research [9] said that the data-driven approach for melody extraction shows better performance than salience-based and source separation-based approaches.

One of the recent studies on this topic is frequency-temporal attention network (FTANet) [4]. This architecture attempts to imitate human hearing to determine the different weights in determining frequency and time in the. Further explanation can be seen in section 3.

There are several query matching techniques, such as Dynamic Time Warping (DTW) in research [10], [11], Hidden Markov Model (HMM) [12], and most recently, Unified Algorithm [2].

DTW algorithm is generally more effective in solving query matching problems and has relatively high accuracy, but this algorithm has a high computational time [11]. In addition, DTW algorithm gave poor results when solving pitch variation problems [10]. HMM-based algorithm has high computational power because it is necessary to conduct training and create models to represent a large music database and doesn't even significantly outperform edit distance method [13].

Unified Algorithm is a combination of Relative Pitch 4-Grams (RP4G), 3-Grams (RP3G), and 2-Grams (RP2G), and Mode Normalized Frequency (MNF). When one algorithm fails to find the desired music, it will be continued with another algorithm. The n-gram algorithm is an algorithm that is easy to implement and has efficient computational time [14]. MNF converts the notes to string representation and then compares the MNF between hum and melody query using edit distance. Edit distance algorithm compares two strings and looks for how many changes need to be made so that the two strings become the same.

III. Proposed Method

The architecture of the proposed QbH system can be seen in Figure 1.

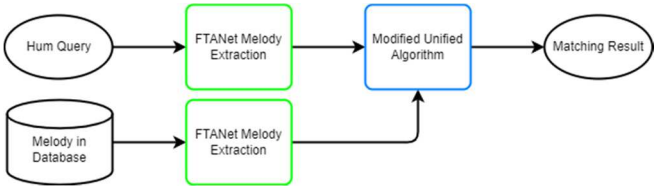

Fig. 1. Architecture of the proposed QbH system. The green box is the module that was purposed to be replaced from the baseline and the blue box is the module that was purposed to be modified from the baseline.

A. Melody Extraction

Melody extraction aims to get the fundamental frequency of the user's hum or melody stored in the database. The melody extraction technique used in this study is Frequency-Temporal Attention Network [4]. FTANet attempts to imitate human hearing by assigning different weights to determine the sound and time frequency. The human auditory system can select a frequency band when sound enters the cochlea.

This network consists of three modules. The first module consists of Frequency attention which attempts to imitate the mechanism of the human auditory system that can select frequency bands when sound enters the cochlea by giving weights to the spectrogram along the frequency axis, and temporal attention, which attempts to imitate the auditory cortex, which performs auto-correlation to obtain temporal relations between frames. The second module is a selective fusion (SF) module to combine spectral and temporal features dynamically. The last module is the melody detection branch (MDB), for detecting the presence of a melody in a sound. Figure 2 shows the frequency-temporal attention network architecture.

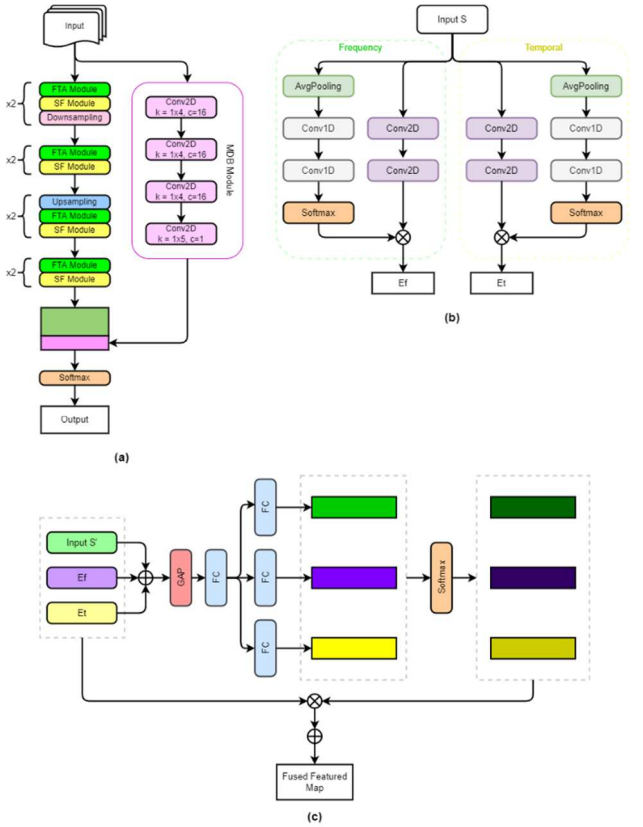

Fig. 2. Frequency-Temporal Attention Network architecture [4]. (a) Main architecture. (b) FTA module architecture. (c) SF module architecture.

Combined Frequency and Periodicity (CPF) is used as input representation because it is effective and popular. CFP is obtained from the calculation of the Short-Time Fourier Transform (STFT) using a sampling rate of 8000Hz, a sample window size of 768, and a sample hop size of 80.

In the FTA module, the distribution of the time axis on the input feature S is calculated using average pooling. Then, frequency-temporal attention is performed using 1-D convolution. The output of 1-D convolution will be forwarded to the softmax layer to get the features of frequency attention A_f and temporal attention A_t. Meanwhile, the input feature S will also enter 2-D convolution layers with kernel sizes (3×3) and (5×5) to generate the new feature $\{S_f, S_t\}$. Then, the matrix

multiplication is processed between $\{A_f, A_t\}$ and $\{S_f, S_t\}$ to get the output $E = \{E_f, E_t\}$.

The output $E = \{E_f, E_t\}$ together with the input S' obtained from the convolution process (1×1) to the input S is used as input to the SF module. A matrix addition will combine these three inputs into one feature, then followed by global average pooling (GAP) and a fully connected (FC) layer for non-linear transformation. In the next layer, there are three FC layers to study the effect of each channel on the features. The process is continued to the softmax layer to get the attention feature, and then this attention feature will be multiplied with the three previous input features to get the feature weight. The weight of each of these features is combined with a matrix addition operation on each of their elements.

MDB consists of four convolutional layers for downsampling the input. The first three convolutional layers have a kernel and stride size (4×1), and the last convolutional layer has a kernel and stride size (5×1). The output of this module will be combined with the output of the selective fusion module [4].

A rough estimate of the minimum duration of a note hummed by an untrained singer is 100 milliseconds [2]. The fundamental frequency that has been obtained needs to be converted into semitone because generally semitone is used more often than the fundamental frequency in the QbH system [15].

Semitone s can be obtained from the fundamental frequency f by the equation:

$$s = 69 + 12\, x\, log_2\left(\frac{f}{440}\right) \quad (1)$$

B. Melody Matching

Melody matching aims to find the melody in the database that has the highest similarity to the user's hum. The semitones obtained from the extraction of the melody will be used in matching the melody between the hum query and the melody query. Each melody in the database has been extracted before, and there is no need to calculate semitones because it is in MIDI format, which already contains information related to the tone.

Unified algorithm was used in this study for melody matching. The n-gram approach is limited to only 4-grams. This is because using higher grams does not always find a suitable melody.

Meanwhile, MNF is an algorithm that performs normalization using alphabetical order as a tone representation with N as the reference for the note that occurs most often, and other notes will be converted based on the relative distance of the note with the reference note.

Unified Algorithm uses the relative distance between notes to eliminate the need for frequency transposition as user hums are usually not perfect exactly like the actual melodies and hum melodies with different octaves. The inverted index, which consists of a list n-grams from relative distance and the number of its occurrences, or the MNF of the melody is used to eliminate the need to calculate it when matching the query.

In addition, the error when there is a difference in tempo between hums and melodies causes the consideration to perform query compression, which eliminates duplicate notes to ignore tempo variations [2].

Often, the song that the user hums is only part of the whole melody. This causes a match to be made between the humming query and the query containing the entire melody. To overcome this, a query containing the entire melody will be taken in part, with the same length as the hum query for melody matching. Figure 3 shows the flow of the original and changes made to the unified algorithm.

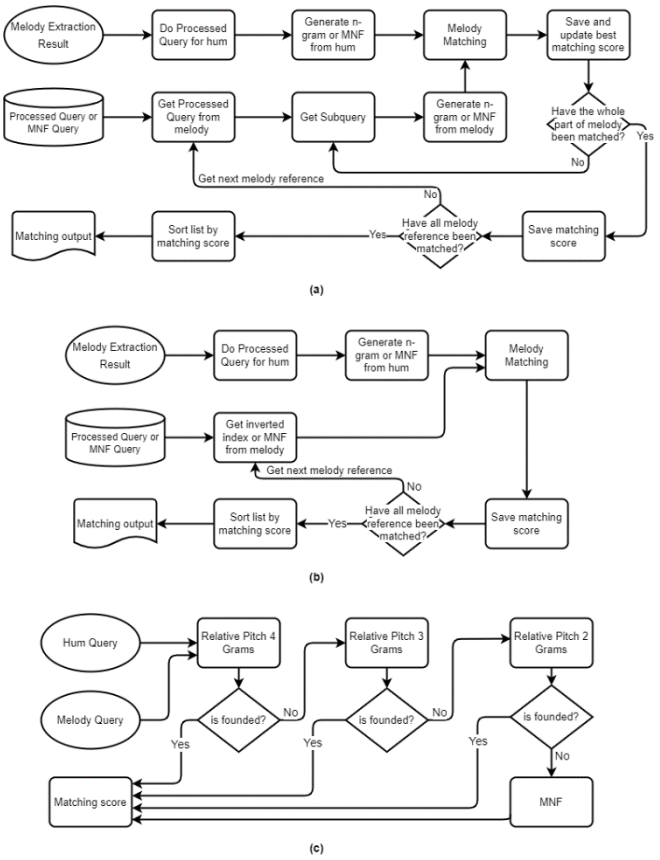

Fig. 3. The flow of Unified Algorithm. (a) Modified Unified Algorithm. (b) Original Unified Algorithm. (c) Details of the matching melody process, where the algorithm is used interchangeably when it doesn't find the right melody reference.

This partial matching is done until the entire query from the melody is matched with the humming query. The matching scores between some of these queries will be compared, and the best score will be taken. The inverted index is not used because it is a list of melodies that have a certain relative distance and are taken from the entire melody. Meanwhile, the query part of the melodies used in this experiment has a dynamic length that adjusts to the length of the query hum. The inverted index is replaced by the result of processing the query, which is ten consecutive frames of the base frequency, converting the fundamental frequency to a semitone, and converting the representation to alphabetical to match the MNF of each melody.

IV. Experiments and Evaluation

A. Dataset

The dataset used in this study is a dataset created for the query by humming task, which was obtained from the Music Information Retrieval Evaluation eXchange (MIREX). This dataset consists of two corpora, namely MIR-QBSH Corpus and IOACAS Corpus. MIR-QBSH Corpus consists of 4431 queries with 48 MIDI files as ground truth. The IOACAS Corpus consists of 759 queries with 298 MIDI files as ground truth.

B. Experiments

Experiments were performed by comparing our proposed method with the state of the art of QbH system that uses autocorrelation as melody extraction and the basic unified algorithm for the matching process [2]. In addition, DTW was also used to perform melody matching using queries from FTANet and compared with the Unified Algorithm.

Compressed and uncompressed queries are used when performing melody matching. A compressed query is a query that removes duplication of tone from the query. Queries from reference melodies are also compressed when performing melody matching.

Evaluation of the QbH system is top n hit ratio, Mean Reciprocal Rank (MRR), and average computational time. The top n hit ratio shows the percentage of reference melodies that enter the top n lists when matching between hum and melodies.

Average computational time $\bar{x}$ from N query with t as time computation from performing process of a stage is described as:

$$\bar{x} = \frac{1}{|N|} \sum_{(i=1)}^{|N|} t_i \quad (2)$$

The MRR equation from Q queries, and $rank_i$ is the rank position of the i-th correct song from the output of the QbH system is described as:

$$MRR = \frac{1}{|Q|} \sum_{(i=1)}^{|Q|} (1/rank_i) \quad (3)$$

Table I, II, III, and IV shows the results of the performance evaluation in each experiment scenario.

TABLE I. Experiment Result from MIR-QBSH Dataset

System	MRR	Top n Hit Ratio			
		1	*3*	*5*	*10*
DTW Uncompressed	0.11	0.04	0.07	0.11	0.23
DTW Compressed	0.12	0.05	0.09	0.12	0.23
Baseline Uncompressed	0.22	0.08	0.21	0.34	0.62
Baseline Compressed	**0.33**	**0.17**	**0.35**	**0.48**	**0.73**
Proposed Uncompressed	0.26	0.11	0.26	0.38	0.68
Proposed Compressed	0.30	0.16	0.30	0.43	0.68

TABLE II. Experiment Result from IOACAS-QBH Dataset

System	MRR	Top n Hit Ratio			
		1	*3*	*5*	*10*
DTW Uncompressed	0.02	0.01	0.02	0.02	0.03
DTW Compressed	0.03	0.01	0.02	0.03	0.04
Baseline Uncompressed	0.06	0.01	0.04	0.06	0.13
Baseline Compressed	0.14	0.07	0.13	0.19	0.28
Proposed Uncompressed	0.08	0.03	0.07	0.10	0.18
Proposed Compressed	**0.20**	**0.11**	**0.20**	**0.27**	**0.38**

TABLE III. Average Computational Time from MIR-QBSH Dataset

System	Time in second	
	Melody Extraction	*Melody Matching*
DTW Uncompressed	2.495	0.525
DTW Compressed	2.495	0.492
Baseline Uncompressed	**0.031**	0.007
Baseline Compressed	**0.031**	**0.005**
Proposed Uncompressed	2.495	1.626
Proposed Compressed	2.495	0.236

TABLE IV. Average Computational Time from IOACAS-QBH Dataset

System	Time in second	
	Melody Extraction	*Melody Matching*
DTW Uncompressed	3.901	10.426
DTW Compressed	3.901	9.921
Baseline Uncompressed	**0.039**	0.052
Baseline Compressed	**0.039**	**0.037**
Proposed Uncompressed	3.901	61.671
Proposed Compressed	3.901	8.475

C. Discussion

Based on the experimental results, it can be seen that the proposed method can provide better performance on the QbH system compared to the baseline system in each scenario except for the MIR-QBSH dataset with compressed queries. However, even though, based on the results of the evaluation, it managed to outperform the baseline system, the average computational time for both melody extraction and melody matching in the proposed method is longer compared to the baseline system. DTW has the lowest score to perform melody matching in all scenarios.

The use of compressed queries on all systems and each dataset has succeeded in increasing MRR and top n hit ratio values and reducing average computation time. In the baseline system, the average increase in computational time was insignificant, 0.002 seconds or 23% in the MIR-QBSH dataset and 0.015 seconds or 29% in the IOACAS-QBH dataset. While on the proposed method, there was a

significant increase of 1.39 seconds or 85%, from 1.626 seconds to 0.236 seconds on the MIR-QBSH dataset, and an increase of 53.196 seconds or 86% from 61.671 seconds to 8.4748 seconds.

Overall, except for the MIR-QBSH dataset with compressed queries scenario, more queries from the proposed method have higher rankings than the baseline system, and based on the results of the MRR and top n hit ratio, the proposed method provides better results than the baseline system.

The proposed method is better than the baseline system in every scenario except the MIR-QBSH dataset scenario with compressed queries because the system provides better melody matching results in the ranking range 1-10, especially in the IOACAS-QBH dataset which has a higher number of ground truth, much more than the MIR-QBSH dataset as shown in Figure 4.

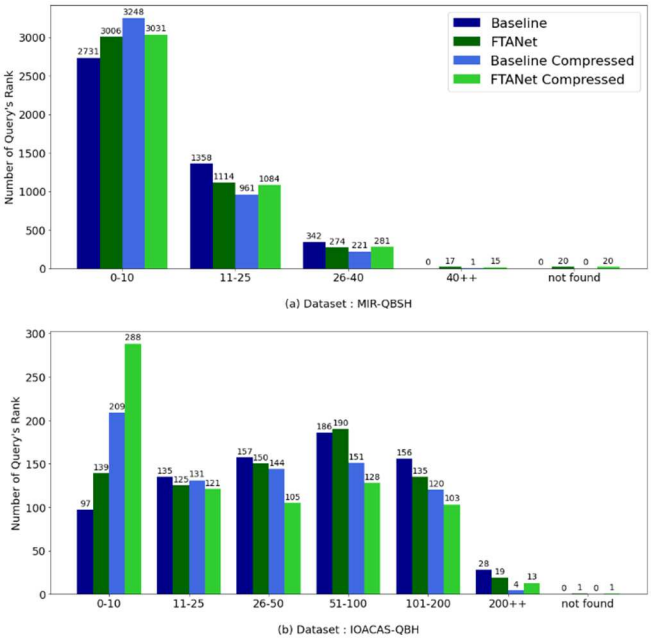

Fig. 4. Comparison of the number of query rankings for both datasets. (a) MIR-QBSH dataset. (b) IOACAS-QBH Dataset

This is because the query from the melody extraction of the proposed method can predict fundamental frequencies that are more similar to the reference query. In addition, partial matching of reference queries also plays a role in improving the evaluation results of the system. By matching the humming query with part of the reference query, the system can match the melody better because the query length of the matched melody is the same. Figure 5 shows examples of queries where the proposed method gives better results.

Furthermore, the baseline system can give better results in the MIR-QBSH dataset scenario with compressed queries because from the extraction results, no reference query matches the query from the extraction of the proposed method, even after partial matching with query reference as shown in Figure 6.

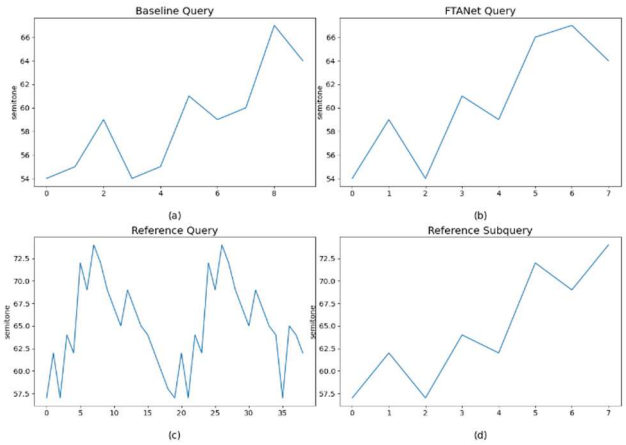

Fig. 5. Examples of queries where the proposed method gives better results. (a) Query from baseline. (b) Query from the proposed method. (c) Query from reference melody. (d) Subquery from reference melody, shows the subquery can provide a more similar representation to the humming query when compared to the whole query from the reference melody.

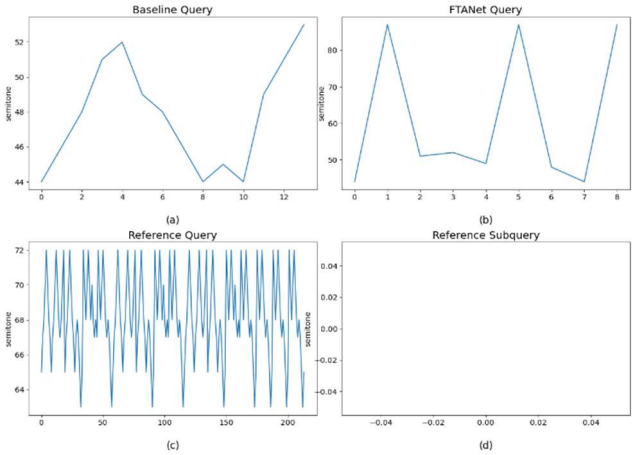

Fig. 6. Examples of queries where baseline gives better results. (a) Query from baseline. (b) Query from the proposed method. (c) Query from reference melody. (d) Subquery from reference melody, shows there is no query from reference melody that matches with the query from the proposed method, even when matched with subquery from reference melody.

In addition, there are 20 queries on the MIR-QBSH dataset and 1 query on the IOACAS-QBH dataset from the melody extraction of the proposed method, as shown in table V, which cannot produce 10 consecutive frames to be used as a query used for melody matching. The melody extraction used by the proposed method is a data-based melody extraction that depends on the training process, and prediction errors can occur. An example of a query that fails to produce 10 consecutive frames is shown in Figure 7.

TABLE V. NUMBER OF QUERIES SUCCESSFULLY REACH MINIMUM DURATION

Melody Extraction	**Dataset**	**Number of queries**	
		Reached min duration	***Failed to reach min duration***
FTANet	MIR-QBSH	4411	20
	IOACAS-QBH	758	1
Baseline	MIR-QBSH	4431	0
	IOACAS-QBH	759	0

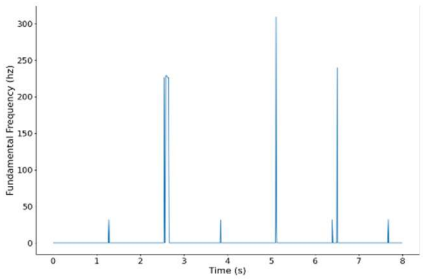

Fig. 7. Example of a query from the proposed method that fails to produce 10 consecutive frames, shows that query FTANet predicts no sound in the frame which results in many timeframes having 0 fundamental frequency.

Although the proposed method succeeded in giving better results than the baseline system, the average computational time of the system was much longer both in the melody extraction process and the melody matching process. Using neural network architecture results in a longer average computational time than the baseline system that uses the autocorrelation method. In addition, adjustments to the unified algorithm to match the melody part also make the computation time much longer.

The use of the Unified algorithm for melody matching is not suitable for datasets with large ground truth. The number of MIDI used as ground truth affects the MRR results. In tables I and II, it can be seen that the experiment on the MIR-QBSH dataset is better than the experiment on the IOACAS-QBH dataset. The reason for this is, that with a large number of MIDI used as ground truth in the IOACAS-QBH dataset, many of the rankings of the searched queries are in the high rankings, as shown in Figure 4.

The number of MIDI and MIDI duration used as ground truth also affect the computational time in matching the melody. The average duration and number of MIDI as a reference in the two datasets have a fairly large difference, as shown in Table VI. The higher the number of MIDIs and the longer the duration of each MIDI, the longer the computation time. This makes the unified algorithm unsuitable for use in systems with large datasets.

TABLE VI. COMPARISON OF THE AVERAGE DURATION OF QUERY AND MIDI FILES AS A REFERENCE IN THE DATASET

Dataset	Avg Duration in second		Number of Queries	Number of MIDI
	Query	*MIDI*		
MIR-QBSH	7.99	39.80	4431	48
IOACAS-QBH	13.13	199.50	759	298

V. CONCLUSIONS AND FUTURE WORKS

The use of FTANet as melody extraction, which is a melody extraction technique with a data-based approach and adjustments to the Unified Algorithm to do partial matching of the melody can provide better performance compared to the baseline system, which uses an autocorrelation-based method for melody extraction even it produces more many queries that have a lower ranking. This can be seen from the evaluation results, both MRR and top n hit ratio. However, in terms of computational time, both melody extraction and melody matching processes take much longer. Then, the Unified algorithm is not suitable for use in systems with large datasets. The number of MIDIs and the duration of MIDI used as ground truth affect the evaluation results and the average computation time.

To improve the performance of the proposed QbH system in this study, the optimization of the matching method to be suitable to use in systems with large datasets is worth a try to provide a much better average computation time.

REFERENCES

[1] A. Ghias, J. Logan, D. Chamberlin, dan B. C. Smith, "Query By Humming : Musical Information Retrieval in an Audio Database," *Proc. third ACM Int. Conf. Multimed.*, hal. 231–236, 1995.

[2] V. Makarand dan K. Parag, "Unified Algorithm for Melodic Music Similarity and Retrieval in Query by Humming," *Intell. Comput. Inf. Commun. Adv. Intell. Syst. Comput.*, vol. 673, 2018, doi: 10.1007/978-981-10-7245-1_37.

[3] P. Boersma, "Accurate Short-Term Analysis of The Fundamental Frequency and The Harmonics-to-Noise Ratio of a Sampled Sound," *Proc. Inst. phonetic Sci.*, vol. 17, hal. 97–110, 1993, doi: 10.1006/jsvi.1998.2072.

[4] S. Yu, X. Sun, Y. Yu, dan W. Li, "Frequency-Temporal Attention Network for Singing Melody Extraction," *IEEE Int. Conf. Acoust. Speech Signal Process.*, hal. 251–255, 2021, doi: 10.1109/ICASSP39728.2021.9413444.

[5] J. Song, S. Y. Bae, dan K. Yoon, "Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System," *ISMIR*, hal. 133–139, 2002.

[6] N. Mostafa dan P. Fung, "A Note Based Query By Humming System using Convolutional Neural Network," *INTERSPEECH*, hal. 3102–3106, 2017, doi: 10.21437/Interspeech.2017-1590.

[7] O. Babacan, T. Drugman, N. D'Alessandro, N. Henrich, dan T. Dutoit, "A comparative study of pitch extraction algorithms on a large variety of singing sounds," *2013 IEEE Int. Conf. Acoust. Speech Signal Process.*, hal. 7815–7819, 2013, doi: 10.1109/icassp.2013.6639185.

[8] J. Abeßer dan M. Muller, "Fundamental frequency contour classification: A comparison between hand-crafted and CNN-based features," *ICASSP 2019-2019 IEEE Int. Conf. Acoust. Speech Signal Process.*, hal. 486–490, 2019, doi: 10.1109/ICASSP.2019.8682252.

[9] R. Kumar, A. Biswas, dan P. Roy, "Melody Extraction from Music: A Comprehensive Study," in *Algorithms for Intelligent Systems*, Springer, 2020, hal. 141–155.

[10] R. A. Putri dan D. P. Lestari, "Music information retrieval using Query-by-humming based on the dynamic time warping," *Int. Conf. Electr. Eng. Informatics*, 2015, doi: 10.1109/ICEEI.2015.7352471.

[11] S. Zhou, Z. Zhao, P. Shi, dan M. Han, "Research on Matching Method in Humming Retrieval," *IEEE 3rd Inf. Technol. Mechatronics Eng. Conf.*, 2017, doi: 10.1109/itoec.2017.8122349.

[12] H.-H. Shih, S. S.Narayanan, dan C.-C. J. Kuo, "Multidimensional humming transcription using a statistical approach for query by humming systems," *IEEE Int. Conf. Acoust. Speech, Signal Process.*, vol. 5, hal. V–541, 2003, doi: 10.1109/ICME.2003.1221329.

[13] A. Kotsifakos, P. Papapetrou, J. Hollmén, D. Gunopulos, dan V. Athitsos, "A survey of query-by-humming similarity methods," *Proc. 5th Int. Conf. PErvasive Technol. Relat. to Assist. Environ.*, hal. 1–4, 2012, doi: 10.1145/2413097.2413104.

[14] K. Gurjar dan Y.-S. Moon, "A comparative analysis of music similarity measures in music information retrieval systems," *J. Inf. Process. Syst.*, vol. 14, no. 1, hal. 32–55, 2018, doi: 10.3745/JIPS.04.0054.

[15] J. Yang, J. Liu, dan W.-Q. Zhang, "A fast query by humming system based on notes," *Elev. Annu. Conf. Int. Speech Commun. Assoc.*, 2010.