A Design of Matching Engine for a Practical Query-by-Singing/Humming System with Polyphonic Recordings

Seok-Pil Lee[1] , Hoon Yoo[1] and Dalwon Jang[2]
[1]Department of Digital Media Technology, Sangmyung University
Seoul, South Korea
[e-mail: esprit@smu.ac.kr]
[2]Korea Electronic Technology Institute
Seoul, South Korea
[e-mail: dalwon@keti.re.kr]
*Corresponding author: Seok-Pil Lee

Received December 20, 2014; accepted January 21, 2014; published February 28, 2014

Abstract

This paper proposes a matching engine for a query-by-singing/humming (QbSH) system with polyphonic music files like MP3 files. The pitch sequences extracted from polyphonic recordings may be distorted. So we use chroma-scale representation, pre-processing, compensation, and asymmetric dynamic time warping to reduce the influence of the distortions. From the experiment with 28 hour music DB, the performance of our QbSH system based on polyphonic database is very promising in comparison with the published QbSH system based on monophonic database. It shows 0.725 in MRR(Mean Reciprocal Rank). Our matching engine can be used for the QbSH system based on MIDI DB also and that performance was verified by MIREX 2011.

***Keywords*:** Query-by-singing/humming, music information retrieval, matching engine, dynamic time warping, pitch sequence, MIREX

A preliminary version of this paper was presented at APIC-IST 2013 and was selected as an outstanding paper.

http://dx.doi.org/10.3837/tiis.2014.02.0024

1. Introduction

With the proliferation of digital music, efficient indexing and retrieval tools are required for searching the desired music in a large digital music database (DB). Music information retrieval (MIR) has been a matter of interest [1,2,3,4]. In the area of MIR, there are some challengeable tasks such as music genre classific, ation [5], melody transcription [6,7,8], cover-song identification [9], tempo estimation [10], audio identification [11,12,13,14,15], query-by-singing/humming (QbSH) [16,17,18,19.20,21,22,23], and so on. Among the various tasks, QbSH is differentiated from other tasks in that it has users' acoustic query as an input. Conventional QbSH systems generally have been developed with DB constructed from MIDI files. Since most QbSH systems use pitch sequences, automatic transcription process is needed for the QbSH system based on polyphonic recordings, but automatically transcribed pitch sequences from polyphonic recordings are not reliable as the sequences from MIDI files [24,25,26,27]. Thus the accuracy of a QbSH system is higher when using MIDI files than polyphonic recordings. However manual transcription is a laborious and time consuming work.

This paper proposes a matching engine for a practical QbSH system with polyphonic recordings like MP3 files as well as monophonic recordings like MIDI files. Various automatic melody transcription algorithms are being developed, but the transcriptions are not yet perfect. **Fig. 1** shows three different processes to obtain pitch sequences and the processes can be thought as channels where the pitch sequence is distorted. Sequence A which is extracted from manually transcribed MIDI files is not distorted, but sequence B and C is distorted by respectively user's process and automatic transcription. The QbSH system based on polyphonic recordings should match sequence B and C while the system based on MIDI files should match sequence A and B.

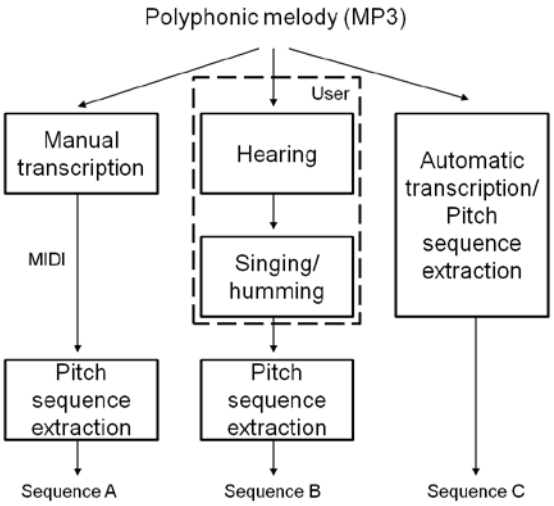

Fig. 1. Three different pitch sequences from a melody: Sequence A is extracted from MIDI files, sequence B is extracted from a user's singing/humming and sequence C is extracted from a polyphonic music.

The main contribution of this paper is a design of a matching engine considering the inaccuracies. We use previously developed algorithms for the pitch transcription [6,7,8]. The matching engine of our QbSH system is based on the asymmetric dynamic time warping (DTW) algorithm [28,29,30]. To reduce the influence of distortions on extracted pitch sequences, we use pre-processing, compensation.

Numerous MIR algorithms and systems are evaluated and compared annually under the controlled conditions from the music information retrieval evaluation exchange (MIREX) contest [1,2,31]. But there is not any contest for the QbSH system based on polyphonic recordings in MIREX. So we compare the performance of the proposed system with the existing system without chroma representation. Also we checked feasibility of our matching engine without chroma representation for the QbSH system based on MIDI files in MIREX 2011 [32].

The rest of this paper is organized as follows. Section 2 presents the overall structure of our QbSH system. Section 3 and 4 explain the two processes of extracting pitch sequences from polyphonic and monophonic recordings and the proposed matching engine, respectively. Section 5 and 6 present experimental results and conclusions, respectively.

2. Overall Structure of the QbSH System

A list of symbols used in this paper is summarized in **Table 1**.

Table 1. A list of symbols

Symbol	Meaning
$\mathbf{S}_q$	Pitch sequence of query
$\boldsymbol{S}^{(i)}_{DB}$	The pitch sequence of the ith song, stored in DB
$ID^{(i)}$	Name of the ith song
I	The number of songs in DB
$L(\bullet)$	Length of pitch sequence
$d_{DTW}(\mathbf{P},\mathbf{Q})$	Distance between two sequences P and Q based on DTW
$d_M(\mathbf{P},\mathbf{Q})$	Distance between two sequences P and Q computed in matching engine
$d(\bullet)$	arbitrary distance metric
α and β	Weighting coefficient for DTW
c	Compensation coefficient
$\psi(\bullet)$	Pre-processing of matching engine

Fig. 2 shows the block diagram of our QbSH system. The processes shown in the left side of **Fig. 2** should be performed offline and the processes shown in the right side of **Fig. 2** are performed with the user's query input. The feature DB is constructed from the MIDI DB or the music DB. The music DB contains a number of polyphonic recordings such as MP3 files. From the data in the DBs, a frame-based pitch sequence

$\boldsymbol{S}^{(i)}_{DB}$ is extracted and the feature DB contains the pitch sequence and the name of music data $ID^{(i)}$ for i = 1, 2, • • • , I. From the user's query, frame-based pitch sequence $\mathbf{S}_q$ is extracted and the matching process based on DTW algorithm [18] finds a DB feature which is matched to $\mathbf{S}_q$. After that, the song name related to the feature is outputted.

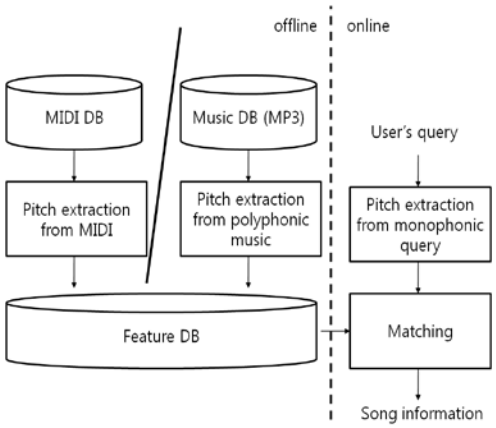

Fig. 2. Overall structure of the QbSH system

3. Pitch Sequence Extraction

In this section, two pitch sequence extraction processes – from polyphonic music and from monophonic query - are briefly explained. A pitch extraction from MIDI is nothing but decoding MIDI. The two processes commonly start with decoding process since input is generally compressed by an audio codec such as MP3. The decoded signal is re-sampled at 8 kHz and framed with a 64 ms frame. For a frame, the frequency of pitch is obtained and converted into MIDI note number domain. Actually the round operator to get an integer number for converting from frequency domain to MIDI note number domain, but a real number is obtained without rounding in our implementation. Mathematically, converting is formulated as

$$M = 12 \times \log_2(f/440) + 69 \tag{1}$$

where M and f are a MIDI number and frequency, respectively [19]. However, the pitch sequence extracted from MIDI has an integer value of MIDI number.

3.1 Pitch Sequence Extraction from Monophonic Query

Since humming query contains uninformative signal such as silence, the voice-activity detection (VAD) algorithm is used for detecting informative frames. Improved minima controlled recursive averaging (IMCRA) algorithm, which is based on the ratio of spectrum of humming and noise signal, is used as a VAD algorithm [28]. The pitch value of 0 is assigned for the non-informative frame. After performing the VAD algorithm, a pitch value of a frame is determined by a method based on the spectro-temporal autocorrelation [29]. The temporal autocorrelation and spectral autocorrelation are computed and linearly combined.

3.2 Pitch Sequence Extraction from Polyphonic Music

A pitch is estimated based on average harmonic structure (AHS) of pitch candidates [30]. From the spectrum of polyphonic music, peaks are selected and the candidates of pitch are determined. Based on the AHS values of candidates and continuity of pitch sequence, a pitch value of a frame is estimated. For polyphonic music, the accuracy of voicing detection is not as reliable as the VAD algorithm for monophonic query, thus a pitch is estimated every frame without voicing detection. The frame is ignored when it is silence. This leads to the temporal distortion of the pitch sequence, but the matching algorithm based on the DTW algorithm is robust against such distortion. It can degrade the performance of QbSH system when the silence lasts very long, but generally silence lasts very short (<<1sec).

3.3 Melody Extraction

The main melody from polyphonic signal is the reference dataset for our query system. Multiple fundamental frequencies as called multi-F0 have to be calculated before estimating main melody from polyphonic music signal which has the various instrument sources plus singer's vocal simultaneously. This topic has been researched by various papers for the last decade, but those articles have informed that estimating multi-F0 is not an easy task; especially when the accompaniment is stronger than main vocal. We can see easily that it happens at the current popular music like dance, rock, something else. Keeping in mind this situation, we propose the method of tracking the main melody from the multi-F0 with the harmonic structure which is very important fact of vocal signal. All of musical instruments have the harmonic structure as well as human vocal but percussion instruments.

The vocal region is detected with the zero crossing rate, frame energy, and the deviation of spectral peaks. We introduce the vocal enhancement module based on the multi frame processing and noise suppression algorithm to improve accuracy of vocal pitch. It is modified from adaptive noise suppression algorithm of IS-127 EVRC speech codec which has the advantage of enhanced performance with relative low complexity. The gain is calculated with SNR between input signal and noise level predicted by pre-determined method at each channel. Input signal is rearranged with this gain at each channel respectively. The noise suppressed input signal is obtained by inverse transformation.

The multi-F0 candidates are estimated from the predominant multiple pitch calculated by the harmonic structure analysis. The multi-F0 is decided by grouping the pitches into several sets by checking validation of its continuity and AHS. The melody is obtained by tracking the estimated F0. Voiced or unvoiced frame is determined on the preprocessing stage. If it is judged to unvoiced frame, it assures that F0 does not exist, otherwise doing harmonic analysis. Multi-F0 is estimated through three processing module like peak picking, F0 detection and harmonic structure grouping. There are some peak combinations with F0 because polyphonic signal is mixed with several musical instrument sources.

4. Matching Engine

4.1 Problems to Solve

A matching engine should be robust against the mismatch between $\mathbf{S}_q$ and $\boldsymbol{S}^{(i)}_{DB}$. As shown in **Fig.1**, two sequences are originated from a melody which can be written in a musical score. But, the information in a musical score which is almost perfectly transmitted to a polyphonic melody may be distorted in the process of extracting pitch sequences. **Fig. 3** shows examples of pitch sequences and explains the tendency of distorted sequence. The pitch sequence from

MIDI is clean and other sequences are distorted. In the following subsections, distortions induced from the processes are reviewed.

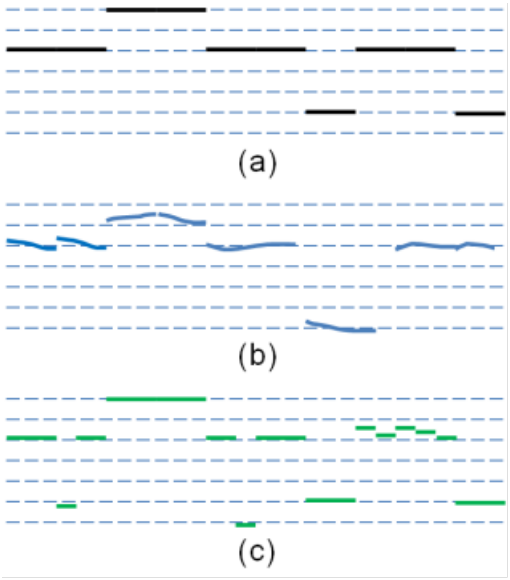

Fig. 3. Example of three different pitch sequences: (a) pitch sequence extracted from MIDI, (b) pitch sequence extracted from user's query, and (c) pitch sequence extracted from polyphonic recording.

4.1.1 Inaccuracy of Singing/Humming

Commonly, users can not sing/hum as in a musical score. Most users sing/hum at inaccurate absolute/relative pitch with a wrong tempo. Sometimes, timing of each note is also wrong. Thus, before the pitch sequence extraction of QbSH system the information written in a musical score is distorted by users who sing/hum. The pitch sequence extraction process also leads to the distortion due to imperfect algorithm, but it is not as severe as the distortion due to the user. As shown in **Fig. 3** (b), pitch values of a note is different from the original value shown in **Fig. 3** (a), and they are unstable. Some notes last longer than other notes. However, user's inaccurate singing/humming leads to the error in both pitch value and timing.

4.1.2 Inaccuracy of Extracting Pitch from Polyphonic Music

Polyphonic music includes main melody mixed with human voices and sounds of various instruments. This makes it difficult to extract the pitch sequence of the main melody. For example, the pitch sequence of accompaniment melody is extracted in the frames where the main melody does not exist. And in some frames where signal of accompaniment is strong, it is very difficult to find the pitch value of main melody even though the signals of main melody exist. Pitch doubling and pitch halving effects also lead to inaccurate pitch extraction. These are unavoidable since pitch extraction is based on detection of periodic signal.

4.2 Outline of Matching Engine

By matching $\mathbf{S}_q$ and $\boldsymbol{S}^{(i)}_{DB}$, the name of a song which is sung/ hummed can be found. The objective of matching engine can be mathematically formulated as follow:

$$\hat{i} = \arg \min_{i=1} d_M \ (Sq, S^{(i)}_{DB}) \tag{2}$$

where it is assumed that $\boldsymbol{S}^{(i)}_{DB}$ for i = 1, 2, • • • , I is stored in DB. How to design the distance computed in matching engine $d_M(\cdot)$ is the main concern of this paper. Information of the $\hat{i}_{th}$ pitch sequence $ID^{(\hat{i})}$ is outputted after deciding $\hat{i}$.

In the matching engine for the QbSH system based on polyphonic music DB, $d_M(\cdot)$ is designed as

$$d_M(\mathbf{S}_q, \boldsymbol{S}^{(i)}_{DB}) = \min_{c \in \mathbf{C}_1} d_{DTW}(\psi(\mathbf{S}_q + c), \psi(\boldsymbol{S}^{(i)}_{DB})) \tag{3}$$

where c and $\psi(\bullet)$ are respectively a compensation coefficient and a pre-processing function, which are explained in the following subsection. In the matching engine for the QbSH system based on MIDI DB, $d_M(\cdot)$ is designed as

$$d_M(\mathbf{S}_q, \boldsymbol{S}^{(i)}_{DB}) = \min_{c \in \mathbf{C}_2} d_{DTW}(\mathbf{S}_q + c, \boldsymbol{S}^{(i)}_{DB}) \tag{4}$$

Fig. 4 shows block diagrams of matching engines.

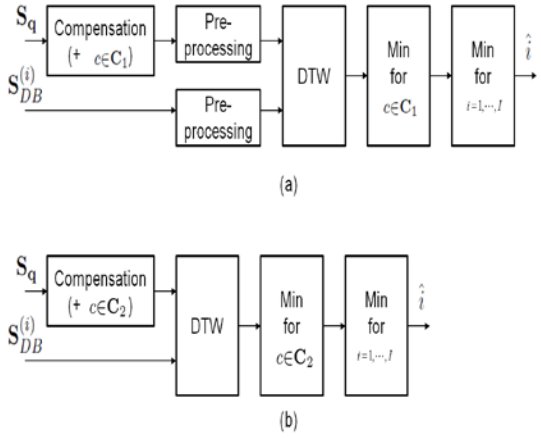

Fig. 4. Block diagrams of matching engines: (a) for the QbSH system based on polyphonic music DB and (b) for the QbSH system based on MIDI DB

4.3 Pre-processing

Pitch doubling and pitch halving effects lead to inaccurate pitch extraction. These are unavoidable since pitch extraction is based on detection of periodic signal. As in [7], the accuracy of raw chroma is much higher than the accuracy of raw pitch. Thus, pitch doubling and halving effects make $\boldsymbol{S}^{(i)}_{DB}$ inaccurate. In the MIDI note number domain, these effects can change pitch values 12 up or down. To avoid these effects, the pitch value is divided by 12 and the remainder is used. $\psi(\bullet)$ is a function to output the remainder of input sequence and the pitch sequence is represented in chroma-scale. For the QbSH system based on MIDI DB, it is better not to use pre-processing since pitch sequence extracted from MIDI is not affected by pitch doubling and halving, the pre-processing cannot improve the performance of QbSH system. Use of $\psi(\bullet)$ leads to use of the cyclic function $\xi(\cdot)$. Assuming that there are two pitch values a and b, which satisfy the values of $\psi(a)$ and $\psi(b)$ are 11 and 0 respectively. The absolute

difference between the two, $|\psi(a)-\psi(b)|$ is 11, but it is reasonable that the difference between the two is 1 since we use the chroma scale. Thus, we introduce $\xi(a) = \min(a, 12 - a)$ and use $\xi(|a - b|)$ instead of $|a - b|$ with the pre-processing in the matching engine for the QbSH system based on polyphonic music DB.

4.4 Compensation

As mentioned in 4.1.1, it is not able to extract accurate absolute pitch from human and that is why we should compensate it. A compensation coefficient c is added to the pitch sequence as written in Equation (4) and the minimum distance is selected. Finding an appropriate c is very difficult because it is difficult to predict the DTW path. Thus, we use a brute-force search which selects a coefficient with minimum distance among candidate values.
Candidate values are selected after consideration of trade-off between performance and speed. For the matching engine of the QbSH system based on polyphonic DB, the range of c is limited to $0 \le c < 12$ because of pre-processing. In our experiment, we used $C_1 = \{0, 1, 2, \cdots, 11\}$ considering the execution time. The more compensations lead to the better results, but the more compensations also lead to the longer computational time. For the matching engine of the QbSH system based on MIDI DB, we used $C_2 = \text{avg}(\boldsymbol{S}^{(i)}_{DB}) - \text{avg}(\mathbf{S}_q) + \{-5,-4,-3, \cdots, 5\}$.

We concentrate on the variation of pitch domain using compensation. This is opposite to use of linear scaling [17], which uses the brute-force search based on the variation of timing. It affects our system to reduce timing error with only DTW algorithm.

4.5 DTW Algorithm

DTW algorithm is widely used for QbSH system since it gives the robust matching results against local timing variation and inaccurate tempo. For ease of explanation, we denote $\mathbf{P} = [P(1)P(2) \cdots P(K)]$ for an arbitrary vector $\mathbf{P}$. We assume $\mathbf{P} = \psi(Sq+c)$, $\mathbf{Q} = \psi(\boldsymbol{S}^{(i)}_{DB})$, $L(\mathbf{S}_q) = K$, and $L(\boldsymbol{S}^{(i)}_{DB}) = L$. For DTW algorithm, we define a distance matrix $\mathbf{D}_1 = [D_1(k, l)]$ and $\mathbf{D}_2 = [D_2(k, l)]$ for $k = 1, 2, \cdots, K$ and $l = 1, 2, \cdots, L$. The Procedure to compute $d_{DTW}(\mathbf{P},\mathbf{Q})$ for two sequences $\mathbf{P}$ and $\mathbf{Q}$ is summarized as follows.

DTW distance computation

Input Sequences are $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{S}_q$.
Do for $k = 1, \cdots, K$ and $l = 1, \cdots, L$

$$D_1(k, l) = \begin{cases} 0 & \text{if } S_q(l) == 0 \\ d(P(k),Q(l)) & \text{otherwise} \end{cases}$$

Do for $l = 1, \cdots, L$
$\mathbf{D}_2(1, l) = \mathbf{D}_1(1, l)$

Do for $k = 2, \cdots, K$
$\mathbf{D}_2(k, 1) = \mathbf{D}_2(k - 1, 1) + \mathbf{D}_1(k, 1)$

Do for $k = 2, \cdots, K$ and $l = 2, \cdots, L$
1. If $k == 2$ or $l == 2$
$\mathbf{D}_2(k, l) = \mathbf{D}_1(k, 1) + \min(\mathbf{D}_2(k, l - 1),$

$\mathbf{D}_2(k-1, l)$, $\mathbf{D}_2(k-1, l-1)$
2. otherwise
$\mathbf{D}_2(k, l) = \min(\mathbf{D}_1(k, l) + \mathbf{D}_2(k-1, l-1),$
$\alpha \times \mathbf{D}1(k, l) + \mathbf{D}2(k-2, l-1),$
$\beta \times \mathbf{D}1(k, l) + \mathbf{D}2(k-1, l-2)$

Output

$$d_{DTW}(\mathbf{P},\mathbf{Q}) = \min_{l=\left[\frac{k}{2}\right], \ldots, 1} \mathbf{D}_2(K, l)$$

As in [18], DTW path is asymmetric and the difference is represented by weighting coefficients α and β. When α = β = 1, the DTW algorithm is symmetric and every path has the same weight. In [18], α = 2 and β = 1 are used, and this is reasonable since the weighted path passing two elements. Actually decision of the value of α and β is dependent on the query dataset. In our system, the distance is not computed for the frames where the pitch is not extracted. As explained in section 3, every element in $\boldsymbol{S}^{(i)}_{DB}$ has a pitch value, thus there is no element of 0. But there can be element of 0 in $\mathbf{S}_q$.

4.6 Distance Metric

The performance of QbSH system is dependent on d(·). In general, the absolute difference or the squared difference is used for QbSH system based on DTW algorithm. As written in Section 4.1.2, the pitch sequences in DB are distorted. The distance metric insensitive to the distortion can perform better. In our works, various distance metrics are investigated together with the absolute difference and the squared difference. They are mathematically formulated as follows:

$$d_{|\cdot|}(a, b) = \xi(|a - b|) \tag{5}$$

$$d_{|\cdot|}{}^2(a, b) = \xi(|a - b|)^2 \tag{6}$$

$$d^{(\lambda)}{}_{HINGE}(a, b) = \begin{cases} \xi(|a - b|) \text{ if } \xi(|a - b|) < \lambda \\ \lambda \quad \text{otherwise} \end{cases} \tag{7}$$

$$d_{LOG}(a, b) = \log(1 + \xi(|a - b|)) \tag{8}$$

$$d_{SIG}(a, b) = \frac{1}{1 + \exp(-\xi(|a - b|))} - 0.5 \tag{9}$$

This paper proposes the last three distances for QbSH system since they are less sensitive to the distortion. The first and second distances are generally used for distance measures and they are also compared in our experiment. The third distance metric limits the maximum value as λ. The fourth distance is based on log-scale and the fifth distance is based on sigmoid function.

5. Experiment

5.1 Polyphonic Case

5.1.1 Experimental Setup

Our DB contains 450 songs amounting to over 28 hours of audio data. The lengths of songs in DB differ from 34 sec to 396sec and the DB covers various genres - Rock, Dance, Carol, Children's song and so on. For test set, 1000 query clips of about 10-12sec which are recorded from 32 volunteers, 14 women and 18 men, are used. In the test set, there are various parts of songs including the introduction and the climax. There are 298 humming query clips and 702 singing query clips in test set. Experiments were run on a desktop computer with i7-2600 CPU and 64bit Window 7. Our implementation is entirely written in C++.

As performance measures, we use the mean reciprocal rank (MRR) over top 20 returns, top 1 hit rate, top 10 hit rate and top 20 hit rate. The MRR is widely used for measuring the accuracy of the QbSH system in MIREX contest [8] and it is calculated by the reciprocal of the rank of the correctly identified song for each query [17].

5.1.2 Results

We tested our system with various combination of α and β. Among them, it gives the best result when α and β are set to 3 and 1 respectively. The experimental results are summarized in **Table 2** with various distance metrics. This result compares favorably to previously reported MRRs for 427 music recordings [19] and 140 music recordings [31] in spite of using polyphonic music. As shown in **Table 2**, $d^{(2)}_{HINGE}(\cdot)$ outperforms among the distances proposed in this paper.

To find out the influence of chroma representation, we test the matching engine without chroma representation and $\xi(\cdot)$. **Table 3** shows the test result. As written in the **Table 3**, use of chroma representation is very helpful for the QbSH system especially based on polyphonic DB.

Table 2. Results of the matching engine for the QbSH system based on polyphonic music DB

$d(\cdot)$.	MRR	Top 1	Top 10	Top 20
$d_{\lvert\cdot\rvert}(\cdot)$.	0.703	0.656	0.803	0.843
$d_{\lvert\cdot\rvert}2(\cdot)$	0.642	0.590	0.748	0.795
$d^{(1)}_{HINGE}(\cdot)$	0.655	0.592	0.763	0.826
$d^{(2)}_{HINGE}(\cdot)$	**0.725**	**0.680**	**0.822**	**0.854**
$d^{(3)}_{HINGE}(\cdot)$	0.718	0.670	0.820	0.853
$d_{LOG}(\cdot)$	0.712	0.667	0.804	0.845
$d_{SIG}(\cdot)$	0.721	0.677	0.817	0.852

Table 3. Results with/without chroma representation and $\xi(\cdot)$

Engine	MRR	Top 1	Top 10	Top 20
Proposed	0.725	0.680	0.822	0.854
w/o chroma and $\xi(\cdot)$	0.644	0.598	0.728	0.764
w/o only $\xi(\cdot)$	0.718	0.671	0.814	0.842

5.2 MIDI Case

We submitted our QbSH system based on MIDI files to MIREX 2011 [32].

5.2.1 Experimental Setup

As presented in MIREX website, we use MRR and top 10 rate as performance measures [31]. For the proposed system, the results based on Roger Jang's corpus, which consisting 4431 8-sec queries and 48 ground-truth MIDI files, are presented. As in MIREX task, 2000 Essen Collection MIDI noise files are added to DB [33]. For the system submitted in MIREX 2011, we present results in MIREX website. There are two different tasks based on MIDI dataset. One is based on Roger Jang's corpus, and it is called Task 1a. The other is based on IOACAS corpus with 106 songs as ground truth and 355 humming queries, and it is called Task 1b.

5.2.2 Results

Table 4 shows the experimental results of the system we submitted to MIREX 2011 and other systems which were submitted in recent 3 years [31]. As written in **Table 4**, our system gives moderate performance for Task 1a, but it gives the best performance for Task 1b.

Table 4. Results of the system submitted to MIREX 2011: comparison to the other systems

System	Year	Task 1a		Task 1b	
		MRR	Top 10	MRR	Top 10
CSJ1	2009	0.91	0.94	0.41	0.43
CSJ2	2009	0.86	0.9	0.8	0.86
HAFR	2009	0.66	0.77	0.68	0.78
HAFR1	2010	0.665	0.771	0.704	0.806
JY1	2010	**0.947**	**0.967**	0.416	0.434
JY2	2010	0.926	0.956	0.469	0.49
YF1	2010	0.915	0.947	0.423	0.445
YF2	2010	0.871	0.912	0.844	0.904
TY1	2011	0.93	0.956	0.437	0.456
TY2	2011	0.881	0.919	0.853	0.899
Proposed	2011	0.897	0.939	**0.912**	**0.941**

6. Conclusions

This paper proposes a practical QbSH system based on polyphonic recordings. When the DB is constructed from polyphonic recordings, it gets easier to create a large DB, but reliability of the data in DB gets worse. Considering the problems originated from pitch sequence extraction, the matching engine proposed in this paper is designed using techniques of chroma-scale representation, compensation, asymmetric dynamic time warping and the saturated distances. From the experiment with 28 hour music DB the results are very promising. Our matching engine can be used for the QbSH system based on MIDI DB also and that performance was verified by MIREX 2011. As a future work, we will focus on the design of more accurate pitch extractor using advanced learning algorithm.

Acknowledgment

This research was supported by a 2013 Research Grant from Sangmyung University.

References

[1] Nicola Orio, "Music Retrieval: A Tutorial and Review," Foundations and Trends in Information Retrieval, vol. 1, no 1, 1-90, 2006. Article(CrossRefLink)
[2] J. Stephen Downie, "The Music Information Retrieval Evaluation eXchange (MIREX) Next Generation Project," project prospectus, 2011.
[3] R. Typke, F. Wiering and R. C. Veltkamp, "A survey of music information retrieval systems," in *Proc. of ISMIR*, pp.153-160, 2005.
[4] G. Tzanetakis, G. Essl and P. Cook, "Automatic musical genre classification of audio signals," in *Proc. of Int. Conf. Music Information Retrieval*, Bloomington, IN, pp. 205-210, 2001.
[5] D. Jang, M. Jin and C. D. Yoo, "Music genre classification using novel features and a weighted voting method," in *Proc. of ICME*, 2008.
[6] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering and R. V. Oostrum, "Using transportation distances for measuring melodic similarity," in *Proc. of Int. Conf. Music Information Retrieval*, pp. 107-114, 2003.
[7] G. Poliner, D. Ellis, A. Ehmann, E. Gomez, S. Streich and B. Ong, "Melody transcription from music audio: Approaches and evaluation," IEEE Trans. on Audio, Speech, Language Processing, vol. 15, no. 4, pp. 1247-1256, 2007. Article(CrossRefLink)
[8] S. Jo and C. D. Yoo, "Melody extraction from polyphonic audio based on particle filter," in *Proc. of ISMIR*, 2010.
[9] D. P.W. Ellis and G. E. Poliner, "Identifying cover songs ith chroma features and dynamic programming beat racking," in *Proc. of Int. Conf. Acoustic, Speech and Signal processing*, Honolulu, HI, 2007.
[10] J. -S. R. Jang and H.-R. Lee, "A general framework of progressive filtering and its application to query by singing/humming," IEEE Trans. on Audio, Speech, and language Processing, vol. 16, no. 2, pp. 350-358, 2008 . Article(CrossRefLink)
[11] J. S. Seo, M. Jin, S. Lee, D. Jang, S. Lee and C. D. Yoo, "Audio fingerprinting based on normalized spectral subband moments", IEEE Signal Processing letters, vol. 13, issue 4, pp. 209-212, 2006. Article(CrossRefLink)
[12] D. Jang, C. D. Yoo, S. Lee, S. Kim and T. Kalker, "Pairwise Boosted Audio Fingerprint," IEEE Trans. on Information Forensics and Security, vol. 4, no. 4, pp. 995-1004, 2009. Article(CrossRefLink)
[13] Y. Liu, K. Cho, H. S. Yun, J. W. Shin and N. S. Kim, "DCT based multiple hashing technique for robust audio finger printing," in *Proc. of ICCASP*, 2009.
[14] P. Cano, E. Batlle, T. Lalker and J. Haitsma, "A review of audio fingerprinting," Journal of VLSI signal processing, vol. 41, no. 3, pp. 271-284, 2005. Article(CrossRefLink)
[15] W. Son, H-T. Cho, K. Yoon and S-P Lee, "Sub-fingerprint masking for a robust audio fingerprinting system in a real-noise environment for portable consumer devices," IEEE Trans. on Consumer Electronics, vol. 56, no. 1, pp. 156-160, 2010. Article(CrossRefLink)
[16] A. Ghias, J Logan and D Chamberlin, "Query by humming: musical information retrieval in an audio database", In Proc. of ACM Multimedia, pp. 231-236, 1995.
[17] L. Wang, S. Huang, S. Hu, J. Liang and B. Xu, "An effective and efficient method for query by humming system based on multi-similarity measurement fusion," in *Proc. of ICALIP*, 2008.
[18] H. M. Yu, W. H. Tsai and H. M. Wang, "A query-by-singing system for retrieving karaoke music," IEEE Trans. on multimedia, vol. 10, no. 8, pp. 1626-1637, 2008.
[19] M. Ryyn¨anen and A. Klapuri, "Query by humming of MIDI and audio using locality sensitive hashing," in *Proc. of ICASSP*, 2008.
[20] X. Wu and M. Li, "A top down approach to melody match in pitch contour for query by humming," in *Proc. of International Symposium of Chinese Spoken Language Processing*, 2006.

[21] K. Kim, K. R. Park, S. J. Park, S. P. Lee and M. Y. Kim, "Robust Query-by-Singing/Humming System against Background Noise Environments," *IEEE Trans. On Consumer Electronics*, vol. 57, no. 2, pp. 720-725, May 2011. Article(CrossRefLink)
[22] J. Song, S. Y. Bae and K. Yoon, "Mid-level music melody representation of polyphonic audio for query by humming system," in *Proc. of Int. Conf. Music Information Retrieval*, 2002.
[23] C. C. Wang, J-S. R. Jang and W. Wang, "An improved query by singing/humming system using melody and lyrics information", in *Proc. of Int. Society for Music Information Retrieval Conf.*, pp. 45-50, 2010.
[24] A. P. Klapuri, "Multiple fundamental frequency estimation based on harmonicity and spectral smoothness," IEEE Trans. on Speech Audio Process., vol. 11, no. 6, pp. 804-816, 2003. Article(CrossRefLink)
[25] C. M. Bishop, *Pattern recognition and machine learning*, Springer, 2006.
[26] S. Schapire and Y. Singer, "Improoved boosting algorithms using confidence-rated predictions," Machine Learning, vol. 37, no. 3, pp. 297-336, 1999. Article(CrossRefLink)
[27] D. Jang, C. D. Yoo and T. Kalker, "Distance metric learning for content identification," IEEE Trans. on Information Forensics and Security, vol. 5, issue. 4, pp932-944, 2010. Article(CrossRefLink)
[28] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," IEEE Trans. on Speech and Audio Processing, vo. 11, pp. 466-475, 2003. Article(CrossRefLink)
[29] Y. D. Cho, M. Y. Kim and S. R. Kim, "A spectrally mixed excitation (SMX) vocoder with robust parameter determination," in *Proc. of ICASSP*, pp. 601-604, 1998.
[30] Z. Duan, Y. Zhang, C. Zhang and Z. Shi, "Unsupervised single-channel music source separation by average harmonic structure modeling," IEEE Trans. on Audio Speech Language Processing, vol. 16, no. 4, pp. 766-778, 2008. Article(CrossRefLink)
[31] MIREX website. http://www.musicir. org/mirex/wiki/MIREX HOME.
[32] D. Jang, S.-P. Lee, "Query by singing/humming system based on the combination of DTW distances for MIREX 2011," http://www.musicir.org/mirex/abstracts/2011/JSSLP1.pdf (2011).
[33] Essen associative code and folk database, http://www.esac-data.org.

Seok-Pil Lee received BS and MS degrees in Electrical Engineering from Yonsei University, Seoul, South Korea, in 1990 and 1992, respectively. In 1997, he earned a PhD degree in Electrical Engineering also at Yonsei University. From 1997 to 2002, he worked as a Senior Research Staff at Daewoo Electronics, Seoul, Korea. From 2002 to 2012, he worked as a head of Digital Media Research Center of Korea Electronics Technology Institute. He worked also as a Research Staff at Georgia Tech., Atlanta, USA from 2010 to 2011. He is currently a Professor at the Dept. of Digital Media Technology, Sangmyung University. His research interests include audio signal processing and multimedia searching

Hoon Yoo received BS, MS, and Ph. D degree in electronic communications engineering from Hanyang University, Seoul, Korea, in 1997, 1999, and 2003 respectively. From 2003 to 2005, he was with Samsung Electronics Co., Ltd., Korea. From 2005 to 2008, as an assistant professor, he was with Dongseo University, Busan, Korea. Since 2008, as an associate professor, he has been with Sangmyung University, Seoul, Korea. His research interests are in the area of multimedia processing.

Dalwon Jang received the B.S., M.S., and Ph.D degrees from Korea Advanced Institute of Science and Technology, in 2002, 2003, and 2010, respectively, all in electrical engineering. He is now with the Smart Media Research Center, Korea, Korea Electronics Technology Institute. His research interests include content identification, music information retrieval, multimedia analysis, and machine learning.