The Music Retrieval Method Based on The Audio Feature Analysis Technique with The Real World Polyphonic Music

Chai-Jong Song, Seok-Pil Lee, Sung-Ju Park, Saim Shin, Dalwon Jang

Digital Media Research Center, KETI, #1599, Sangam-dong, Mapo-gu, Seoul, South Korea

ABSTRACT

This paper describes a method of the Music Retrieval Method based on audio feature analysis techniques. This method contains three major algorithms with newly proposed advanced way and the implementation of the whole system including client and server side prototype to be applied on time to market. The first one of the major algorithms is to extract the feature from the polyphonic music, which is the advanced version using the harmonic structure of vocal and musical instruments. The second one is to extract the feature and suppress the noise of user humming signal recorded from input device. Noise suppression algorithm makes merge of MS for stationary noise and IMCRA for non-stationary noise, and the feature is estimated with temporal and spectral autocorrelation simultaneously to reduce the pitch having and doubling problem. The last one is the fusion matching engine improved with DTW (Dynamic Time Warp), LS (Linear Scaling) and QBCode (Quantized Binary Code). This system is extremely targeting on industrial services such as music portal service, fixed stand-alone devices, mobile devices, and so on. Especially, our very first focus is the Korean KARAOKE system which is the one of very popular music entertainment services in Asia and the music portal service like Bugs music, Mnet, zillernet, and so on. We have cooperated with TJ media co. to commercialize this system.

Keywords: MIR, QbSH, Multi-F0, Melody extraction, Pitch contour, Matching engine, DTW, LS, QBcode.

1. INTRODUCTION

With the recent proliferation of digital contents, there are increasing demands for efficient management of the large digital contents’ databases, and the tag based retrievals have been extensively used. But, the way of tagging manually is a laborious and time-consuming work. It has been reported that more than 40,000 albums are released in a year just for only USA music domain. To avoid such works, MIR (Music Information Retrieval) techniques are emerging rapidly rather as we thought as an alternative way to manage a music database [1]. MIREX (Music Information Retrieval Evaluation eXchange) suggested by J. Stephen Downie, who is professor of University of Illinois, have given an impetus to developing MIR techniques for recent years. It has been held every year from 2005, and a lot of participants have competed with other teams having their At the client side device, recording user humming signal with background noise for 10 seconds and then suppressing noise, extracting melody from this signal and finally it

own algorithms or systems[2]. Among various tasks in MIREX, QbSH (Query by Singing/Humming), which provides the music retrieval service to user who only knows some pieces of melody but nothing else, is started with the beginning of that contest[2]. In a few recent years, the applications related music are showing a steady growth with exploding smart phones and tablet users triggered by iphone. There are two popular music retrieval services commercially such as Soundhound and Shazam as you already know. Shazam has served music retrieval based on fingerprinting but QbSH. This is out of the focus of this paper, so we do not consider music fingerprinting anymore. Soundhound taken from online Midomi service has provided QbSH service only with humming feature database which is extracted from user humming to search music in advance. Like this, current QbSH method has been studied with the monophononic signal such as humming or MIDI. But, there are some problems for this method to provide the commercial service: Data sparseness when using humming database, additional works to transcribe the music manually when using MIDI database, and so on. It is difficult to adopt QbSH service for the various industrial fields in case of only targeting monophononic data. We propose the music retrieval method with polyphonic music such as MP3 to eliminate those problems. We are explaining the proposed method starting from briefly description of overall architecture.

2. OVERALL ARCHITECTURE

In this paper, we introduce the proposed music retrieval method with feature analysis technique. There are two main parts for description, the system implementation and the three proposed algorithms. The system implementation contains the client prototype for PC and mobile phone application of Android and server side prototype. The first one of the three major algorithms is to extract features from polyphonic music recordings, the next one is noise suppression and pitch extraction from user humming and the last one is matching algorithm to evaluate the similarity between two of those features. We are considering the three kinds of features: melody, rhythm and segmented section. We are only utilizing the melody and segmentation but not yet rhythm so far. The whole system is operated as followed. transmits the query data formatted with MP-QF international standard to the server waiting for request as you expected. You can switch the noise suppression block turning on and
off by the background noise circumstance. The server parses the received data and calculates the similarity score between queried data and features stored in database, and then recommends top 20 items with highest similarity score to client vice versa. There are three kinds of database of our system as called polyphonic, humming and segmentation. The polyphonic feature is mainly used to make the matching engine evaluate the similarity. The segmentation is to speed up matching algorithm by pre-clustering specific section of music structure such as intro and climax. For the standalone device, server and client function become one whole block without formatting query data.

We have three steps to realize our system. At the beginning of the project, we have started with Roger Jang’s corpus DB used for Q6SH task of MIREX 2005 [3]. It has 8kHz sampling rate and 8bits per sample. We make the very first algorithms of pitch estimation for humming and matching engine get verified with this corpus which have the manuscript pitch vectors represented with semitones at every 32ms. Semitone is represented as Eq. (1).

\[
\text{Semitone} = 12 \times \log_2 \left( \frac{F_0}{440} \right) + 69
\]  

(1)

Here, F0 is fundamental frequency. At this stage, we develop the noise suppression algorithm for humming data with Aurora2 dataset which has stationary and non-stationary background noise including several categories of car, airport, subway, babble, restaurant, train, exhibition and street on the real circumstance. The noise level is settled to 10dB which is similar to real world humming situation. At the next stage, the matching engine is improve the performance with MIDI dataset and 1,200 humming clips recorded for pitch estimation. At the final stage, we have optimized matching engine and feature extraction algorithm with polyphonic music data.

### 3. PITCH ESTIMATION

As I mentioned briefly, pitch estimation algorithm is started with Roger Jang’s corpus and Aurora2 noise dataset for user humming signal, and then we make it suitable to 1,200 humming dataset. Overall procedure for pitch estimation is described as Figure 2. The input signal is sampled at 8kHz with 16bits per sample. It is processed with 32ms frame size and 16ms hoping size. The autocorrelation is the most well-known method for finding pitch from periodic signal and also robust against the noise. It is the powerful tool but it is also the obvious fact that it also has the critical problem for pitch estimation.

\[
R_t(\tau) = \frac{\sum_{n=0}^{N_t-1} x[n] x[n+\tau]}{\sqrt{\sum_{n=0}^{N_t-1} x^2[n] \sum_{n=0}^{N_t-1} x^2[n+\tau]}}
\]

(2)

Here, x[n], \tau, N_t is input signal, delay, and frame length respectively. The spectral autocorrelation is shown as Eq. (3).

\[
R_w(\tau) = \frac{\sum_{k=0}^{N_w/2-1} X[k] X[k+\tau]}{\sum_{k=0}^{N_w/2-1} X^2[k] \sum_{k=0}^{N_w/2-1} X^2[k+\tau]}
\]

(3)

Here, X[k], \tau, N_t is log magnitude spectrum, delay and FFT length of each frame. R_t(\tau) and R_w(\tau) is merged with different ratio after normalized with each energy. Merged autocorrelation depends on the weighting factor \( \beta \) settled at 0.5 through the various experiments. It shows the result that \( \beta \) is better less than 0.5 for woman, vice verse for man. Merging method of each autocorrelation is shown as Eq. (4).

\[
R_\ell(\sigma) = \beta R_t(\sigma) + (1 - \beta) R_w(\sigma)
\]

(4)

At first, we estimate the some pitch candidates with the peak indexes of temporal autocorrelation, and take linear interpolation with only some indexes of spectral autocorrelation in contiguity with candidate pitches of time domain. Linear interpolation on frequency domain is shown as Eq. (5).

\[
R_w(\tilde{\tau}) = R_w(\tau_1) + \frac{R_w(\tau_2) - R_w(\tau_1)}{\tau_2 - \tau_1}(\tilde{\tau} - \tau_1)
\]

(5)
Before spectral autocorrelation, the formant structure is flattened by whitening spectrum because the harmonic structure is broken easily at the high frequency. So we take the spectral autocorrelation after giving salience to the low frequency. Before estimating the pitch candidates, VAD (Voice Active Detection) module detects the voiced frame with high frame energy and low ZCR (Zero Crossing Rate) to extract correct pitch from noise humming data. Once judged to voiced frame, that is considered whether tainted by noise or not. If it is, noise suppression algorithm optimized for our QbSH system is activated for the noisy input. It takes the spectral magnitude of the noisy humming signal through the FFT analysis, and estimates the noise using MS (Minimum Statistics) which assumes that the tainted frames have the minimum power from the noisy signal and IMCRA (Improved Minima Controlled Recursive Averaging) which uses SNR of the statistic ratio between the voiced region and the unvoiced region [5]. Noise suppressed signal is calculated by taking IFFT. At the post-processing stage, the pitches as assumed shot noise are eliminated with median filter.

**Pre-processing:** The music signal on the music database is sampled at 44.1 kHz with 16 bits per sample at stereo. This is down-sampled at 8 kHz and mono before pre-processing to emphasize the pitch information. It is processed by 16ms frame length with Hanning window and one frame look-ahead. 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. Windowed signal is transformed into frequency domain with STFT, and then grouping frequency signal $X[k]$ into 16 channels. 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. This paper assumes the input signal as “vocal melody + accompaniment” while EVRC assumes the input signal as “voice + background noise”. This method improves up to maximum 10.7% accuracy rate for the melody extraction.

**Multi-F0 Estimation:** 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(average harmonic structure). The melody is obtained by tracking the estimated F0. Voiced or unvoiced frame is determined on the pre-processing 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. F0 with several harmonic peaks is evaluated by Eq. (6).

$$|X[k]| > |X[k - 1]| \text{ and } |X[k]| > |X[k + 1]| \text{ and }$$

$$|X[k]| \text{PTH}_{1h}^\text{low}$$

Here, $\text{PTH}_{1h}$ is low and high band threshold for peaks. In general, average energy is different between two bands for the music signal, we make the point at 2kHz to split into two bands. $\text{PTH}$ is adaptively decided by skewness of frequency envelop.

$$SK = \sum_{k=0}^{N/2} (|X[k]| - \bar{X})^2$$

Here, SK is skewness and $\bar{X}$ is mean of $|X[k]|$. If SK=0 energy is symmetric, if SK>0 energy is leaned to low band, if SK<0 high band has the more energy than low band. So, if $SK=0$ then $\text{PTH}_{1l} = \text{PTH}_{1h} = \bar{X}$ and if $SK<0$ then $\text{PTH}_{1l} = \bar{X} - \sigma_d, \text{PTH}_{1h} = \bar{X} + \sigma_d/2$, and if $SK>0$ then $\text{PTH}_{1l} = \bar{X} - \sigma_d/2, \text{PTH}_{1h} = \bar{X} + \sigma_h$. Where, $\bar{X}, \bar{X}, \sigma_d, \sigma_h$ is mean value and standard deviation

4. 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 [6][7][8][9]. You 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 proposed method is shown as Figure 3.

![Melody extraction flow gram](image_url)

**Figure 3. Melody extraction flow gram**
for full band and high band respectively. F0 is limited from 150Hz to 1kHz. The distance of each peak is calculated by Eq. (8).

\[ \Delta[u, v] = \text{peak}[u] - \text{peak}[v] \]  

(8)

Here, \( u=v+1, \ldots, J \), \( v=1, \ldots, J \), \( J \) is total peak number for current frame. The harmonic relation is calculated between peak[v] and every F0 candidates.

**Vocal melody extraction:** If all of the F0 satisfies the ideal harmonic structure, real frequency peak will be at the harmonic peak which they must be. Following this process, you can take 5 F0 candidates at 150, 200, 300, 400, and 450. F0 is assumed as the maximum spectrum peak. AHS (Average Harmonic Structure) determine F0 significant degree by calculating the average energy of harmonic peaks. Vocal melody is tracking estimated F0 candidates of each frame.

**Segmentation:** Rhythmic feature including the tempo is defined by fluctuation pattern of the music clip. In the strict sense of the word, segmentation is not a feature for directly matching process. That is actually one of the pre-processing to enhance matching process by marking at the specific region of music. The modern popular music can be divide into 5 sections as intro, outro, verse, bridge and refrain or chorus. The musical structure analysis method is developed by several studies, but it is out of focus. The focus of this paper is finding the phrase of music which is hummed more often by users. To make it successfully, we utilize the fact that the most modern western pop music has the repeat parts on rhythm and lyrics. Some papers provide the music thumbnail using this feature because it is able to find the climax or interesting part with this fact [10] [11]. Many low level audio features are reported for the segmentation processing like MFCC (Mel Frequency Cepstral Coefficient), chroma, key, fluctuation, energy, ZCR, and so on. We will get the chroma vector which is commonly known as one of the most suitable features for musical structure analysis because it is not dependent on the timbre of particular musical instruments or vocal sound while MFCC is [12]. The segmentation is realized by combining analysis and reconstruction of the audio data. Input audio signal has 20kHz sampling rate, 16 bits per sample, then the STFT with 4096 sample Hanning window is calculated by FFT with the 50 % overlap hop size. Segmentation is realized by following procedure. At first, 12 dimensional chroma vector is calculated from the magnitude spectrum with log-scale band pass filter.

\[ V_t[c] = \sum_{k \in Sc} \frac{X_t[k]}{N_c}, c = \{0, \ldots, 11\} \]  

(9)

Here, \( V_t[c] \) is chroma vector for t-th frame, Sc is chroma set for each octave, \( X_t[k] \) is magnitude spectrum and k is its index. Each element of chroma vector represent one of the pitch classes respectively such as C, C#, D, D#, E, F, F#, G, G#, A, A# and B. It is sum of all values of pitch classes over 6 octaves corresponding from 3 to 8. Then the similarity matrix is calculated by normalized Euclidian distance with chroma vector against time lag.

\[ S(t, l) = \frac{|V_t[c] - V_t[c-l]|}{\sqrt{2}} \]  

(10)

\( S(t,l) \) is satisfied \( 0 \leq S(t,l) \leq 1 \). It shows the repeated section with high score along the horizontal line. The threshold for repeated section is calculated by an automatic threshold selection method based on a discriminant criterion [13]. You can find the optimal threshold by maximizing total variance between two classes.

\[ \alpha_t^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2 \]  

(11)

Here, \( \omega_0 \), \( \omega_1 \) is probabilities of class and \( \mu_0 \), \( \mu_1 \) is mean of peaks in each class.

## 5. MATCHING

The matching engine measures the similarity between pitch contour from humming and melody contour from the music. It returns the top 20 candidates with higher score from fusion matching method which is proposed by this paper. This proposed method is taken and improved from three kinds of algorithm, DTW (Dynamic Time Warping), LS (Linear Scaling) and QB (Quantized Binary) code [14]. It is starting with eliminating the silent duration from pitch and melody contour since it does not have any information for measuring the similarity. Then it normalizes two contours as test and reference vector through Mean-shifting, Median and Average filtering, Min-max scaling.

![Matching engine flow gram](image)

It is why doing Mean-shift filtering that each humming might be located at higher or lower notes rather than original version of music. It has to eliminate the difference and adjust the level of test and reference vector. Median & Average filtering with 5-tap is adopted to remove the peak point caused by surround noise, shivering or vibration of sound tone. Min-max scaling is applied to compensate the distance of amplitude between two vector sequences. After normalize the two vectors, three algorithms calculate the similarity...
scores simultaneously, then those scores are combined with weighting into single fusion score that is the main fact to determine the candidates.

**Dynamic Time Warping:** The major one of three algorithms is the improved DTW which is the popular one of Dynamic Programming to measure the distance between two patterns with different length. Conventional DTW has the several important constraints such as alignment of start and end point, local region constraint, and something else in common, but proposed DTW does not have any constraints. In addition, it calculates the distance between two vectors with log scale to choose the one that has the more elements with small distance, if the distance of similarity is the same. Here is an example. Test vector is \([1,2,1,0,-1]\), and reference vectors are \([2,1,-1,0,4]\) and \([4,5,3,0,2]\). It is the same distance for two reference vector with conventional method. You can select the first one with log scale distance measurement. We can take the improvement by removing the alignment of start and end point because it is able to increase the possibility of matching the start point and length of sequence between two vectors.

**Quantized Binary code:** QBcode has the 4 section of normalized vector and different binary codes are assigned to each section as ‘000’, ‘001’, ‘011’ and ‘111’. The distance for QBcode is calculated with humming distance (HD) between two vectors. We decide whether applying DTW or not with this humming distance.

**Linear Scaling:** LS algorithm is the simplest and quite effective one for patterns which have the different length. The main idea is rescaling test vector into several different lengths for the reference vector. Especially humming length depends on who is humming. So, humming data should be compressed or stretched to match with reference data. Test vector are rescale by scale factor from x1.0 to x2.0 with 5 steps. The distance is measured with the log scale for the same reason of DTW.

**Score level fusion:** The three scores from the above different matching algorithms are merged into the one fusion score. There are many methods for score level fusion as MIN rule, MAX rule, SUM rule and so on. The fusion score is calculated with the PRODUCT rule which multiply two scores. Basically, Proposed DTW carries out the most important role on the matching stage, and LS and QBCode is complement for DTW. So it gives the weight as 0.5, 0.2 and 0.3 to DTW, LS and QBCode respectively. The matching engine recommends top 20 candidates with higher fusion scores.

### Datasets

The reference dataset contains vector sequence and segmentation from 2,000 MP3. Humming test set is consisted of 1,200 humming vector sequences against 100 songs as called AFA100 which is among 2,000 songs referred to MNet music chart that is the most popular one of Korean music portal services. We also have 2,000 MIDI data from KARAOKE system to verify our matching algorithm. We include this MIDI data into our system for Korean KARAOKE service at implementation phase. The music dataset covers 7 different genres with ballad, dance, children song, carol, R&B (rhythm and blues), rock, trot and well-known American pop. The 1,200 humming clips with 12 second duration are recorded against AFA100 to evaluate the algorithms because it is hard to hum at every time of testing the algorithms. It is consisted with almost same ratio of sing and humming and recorded from 29 persons. Three among them have the experience of music related study at university and others not. We analysis and classify that into 3 groups as beginning, climax part and others. We figure out that beginning part is a slight over 60% and climax part is about 30%. It did not expect that the beginning part is almost twice of climax. We evaluate the performance with this humming set.

**Evaluation:** we have evaluated above three algorithms as pitch extraction from user humming, melody extraction from polyphonic music and matching algorithm between test and reference vectors. The evaluation for the pitch extraction algorithm is shown as Table 1. We choose G.729 and YIN as well-known algorithm for pitch estimation for evaluating proposed method [4][15].

<table>
<thead>
<tr>
<th>Environment</th>
<th>G.729</th>
<th>YIN</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear</td>
<td>95.7</td>
<td>97.3</td>
<td>97.3</td>
</tr>
<tr>
<td>babble</td>
<td>90</td>
<td>81.4</td>
<td>88</td>
</tr>
<tr>
<td>wave</td>
<td>93.6</td>
<td>95.7</td>
<td>97.4</td>
</tr>
<tr>
<td>white</td>
<td>93.3</td>
<td>90.1</td>
<td>91.6</td>
</tr>
<tr>
<td>total average</td>
<td>96.4</td>
<td>86.8</td>
<td>90.1</td>
</tr>
</tbody>
</table>

Table 1. Evaluation with GER-10%

The second evaluation is the melody extraction algorithm with two methods as MMR used on TREC Q&A and RPA and RCA used on MIREX contest for melody extraction task. MMR is defined as Eq. (12).

\[
MMR = \frac{1}{N} \sum_{n=1}^{N} \frac{1}{\text{rank}_n} 
\]

(12)

Here, \(N\) is total number of frames, \(\text{rank}_n\) is F0 rank of n-th frame. We have the tolerance of 1/4 tone. We take 0.86 average of MMR. We take the ADC 2004 dataset for evaluating the algorithm because the Korean dataset we have does not have the groudtruth [16]. The last one is the matching algorithm evaluation with MMR method for the recorded 1,200 humming clips. We evaluate the two input
steps as 32ms and 64ms. The evaluation condition as followed: 1,200 humming clips for test vector, 2,000 polyphonic songs with from 3 to 6 minutes duration for reference vector on Intel i7 973 with 8MB memory.

<table>
<thead>
<tr>
<th>Participant</th>
<th>CPA(%)</th>
<th>RCA(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cao and Li</td>
<td>85.625</td>
<td>86.205</td>
</tr>
<tr>
<td>Durrant &amp; Ichord</td>
<td>86.96</td>
<td>87.398</td>
</tr>
<tr>
<td>Hoa. Jiang &amp; hen</td>
<td>83.11</td>
<td>74.101</td>
</tr>
<tr>
<td>Joo. Jo &amp; Yoo</td>
<td>81.959</td>
<td>85.788</td>
</tr>
<tr>
<td>Dressler</td>
<td>85.969</td>
<td>86.424</td>
</tr>
<tr>
<td>Wendelnboe</td>
<td>83.135</td>
<td>86.593</td>
</tr>
<tr>
<td>Cancelo</td>
<td>86.062</td>
<td>87.545</td>
</tr>
<tr>
<td>Rao and Rao</td>
<td>81.446</td>
<td>88.638</td>
</tr>
<tr>
<td>Ono &amp; Sugiyama</td>
<td>89.568</td>
<td>72.129</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>90.418</td>
<td>92.277</td>
</tr>
</tbody>
</table>

Table 2. MIREX 2009 melody extraction result

<table>
<thead>
<tr>
<th>Time</th>
<th>Top1@20</th>
<th>Top2@20</th>
<th>Top3@20</th>
<th>SRER</th>
<th>Trim(Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>32ms</td>
<td>92.10%</td>
<td>90.90%</td>
<td>89.30%</td>
<td>0.768</td>
<td>7.4</td>
</tr>
<tr>
<td>64ms</td>
<td>81.10%</td>
<td>80.10%</td>
<td>79.70%</td>
<td>0.738</td>
<td>7.4</td>
</tr>
</tbody>
</table>

Table 3. Evaluation of matching engine

Implementation: We have implemented three kinds of prototype agents for pitch extraction, melody extraction and matching engine. We also implement two kinds of application for commercial service as server/client version for PC and Android client application. The client application records the humming through the input device as microphone, then extract and send the pitch sequence to the server side and then waiting for response of server. The server is waiting for query after initializing the feature DB, then starting the matching process with receiving query from client. It sends the top 20 candidates with metadata including title, singer, cropped lyrics, genre parsed from MP3 tags to the client. The client chooses the one of recommends, server starts the streaming of data. If it is the finding one, the client send the number of selected item, then server adds the queried humming pitch to humming DB and update the feature DB status and increase the priority of that song. The database structure is built by the technique of the multi dimensional index which provides the efficient search.

7. CONCLUSION

In this paper, we proposed three new algorithms of pitch extraction, melody extraction and matching algorithm for QbSH system with real world polyphonic music. For the pitch extraction algorithm, we propose new idea with merging time and frequency domain autocorrelation with salience interpolation to remove the pitch halving at high frequency and the pitch doubling at low frequency. On the melody extraction stage, which is the most important part for the QbSH with polyphonic music, the algorithm based on the harmonic structure is proposed and segmentation for finding the intro and the climax section is proposed. The last algorithm for the matching engine utilizing score level fusion with three different algorithm as DTW, LS and Qbcode is proposed. Finally, we implement the application for PC and smart phone. In future works, we have the plan to enhance the matching accuracy and implement DSP for embedded system.

8. REFERENCES