# Mid-Level Music Melody Representation of Polyphonic Audio for Query-by-Humming System

Jungmin Song
LG Electronics
16 Woomyeon-Dong, Seocho-Gu
Seoul, Korea
+82-2-526-4111
jmsong73@lge.com

So Young Bae
LG Electronics
16 Woomyeon-Dong, Seocho-Gu
Seoul, Korea
+82-2-526-4111
sybae@lge.com

Kyoungro Yoon
LG Electronics
16 Woomyeon-Dong, Seocho-Gu
Seoul, Korea
+82-2-526-4133
yoonk@lge.com

## ABSTRACT

Recently a great attention is paid to content-based multimedia retrieval that enables users to find and locate audio-visual materials according to the intrinsic characteristics of the target. Query-by-humming (QBH) is also an application that makes retrieval based on major characteristics of music, that is, "melody". There have been some researches on QBH system, most of which are to retrieve music from symbolic music data by humming query. However, when the usability of technology is taken into consideration, retrieval of music in the form of polyphonic raw audio would be more useful and needed in the applications such as internet music search or music juke box, where the music data is stored not in symbolic form but in raw digital audio signal because such music data is more natural format for consumption. Our focus is on the realization of query-by-humming technology for an easy-to-use application, which entails full automation of all the processes of the system, including melody information extraction from polyphonic raw audio. In our system, melody feature of music database and humming is not represented by distinct note information but by the probability of note occurrence. Similarity is then measured between the melody features of humming and music data using DP matching method. This paper presents developed algorithms and experimental results for key steps of QBH system including the melody feature extraction method from polyphonic audio and humming, their representation for matching, and matching method between represented melody information from polyphonic audio and humming.

## 1. INTRODUCTION

Music information retrieval is a research area of developing technologies, which helps retrieving music data to accomplish the process of production, acquisition, transmission, and consumption of music in easy and comfortable way. Query-by-humming (QBH) is one of the music retrieval methods, which finds music that contains similar or same melody to the humming query. Humming a tune is very natural and familiar habit of human and humming as a query for retrieval would be also very natural and convenient way for finding music of interest among a group of music. Recognizing this fact, several technologies have been developed to accomplish this function, and they reported quite promising results in terms of their retrieval accuracy and speed. But most of those technologies are targeted to match humming queries to symbolic music [1][2]. The assumption made by these works is that pitch information or melody information has been extracted from music or target music is in symbolic format (i.e. MIDI files). Approximate matching method is the major choice of these systems, allowing errors or partial variations of the humming query. Allowing errors or partial variations in QBH system is very important as most of the humming by users have errors intrinsically [3][4].

However, the natural music format for most of the music consumers is not a symbolic music, but polyphonic audio that can be obtained from CD's or decoded MPEG audio files. If we want to include a song available through CD or MPEG compressed files in the symbol-based QBH system, the song should be represented in symbols used in the selected QBH system. There are two possibilities in converting raw polyphonic audio into symbol representation. One is the manual conversion, which requires tremendous and tedious human work of extracting melody information. The other is using automatic conversion engine, but it is well known that melody extraction from polyphonic raw audio is a hard-to-solve problem. Though some researchers are trying to solve this problem using various techniques [5][6][7], it seems that they cannot guarantee the accuracy of the melody extracted from the generic music signal with large number of sound mixture, and so does the performance of the QBH system utilizing the approximate symbol matching method based on the extracted melody information.

The QBH system proposed in this paper starts from an effort to avoid the hard-to-solve problem of extracting exact pitch information or melody information from polyphonic audio in signal level, which is also known as automatic transcription problem. In the proposed system, melody feature information is extracted at mid-level, which is conceptually lower than the symbolic representation of melody, but higher than the raw audio signal. To be specific, the melody information is represented by the sequence of vectors whose element describes the estimated strength of note in audio signal, expressing probability of each note being a melody element. Main focuses of this approach are how to extract and represent melody information with the level of accuracy to enable the matching of humming query and the music data, from polyphonic audio, and how to measure the similarity between melody representations.

The melody information extraction and representation method and matching method for matching music and humming query in the proposed mid-level representation are presented in this paper. Experimental results and some discussion on this approach are also presented in the last part of the paper.

## 2. SYSTEM OVERVIEW

Query by humming system is composed of three main modules, which are music melody feature extraction module, humming melody feature extraction module and similarity measuring module as shown in Figure 1. The music melody feature extraction module extracts melody information from music and stores extracted melody information in the melody feature database. Music melody feature is represented as a sequence of vectors, each element of which indicates the probability or the strength of note occurrence of audio frame or audio segment. The

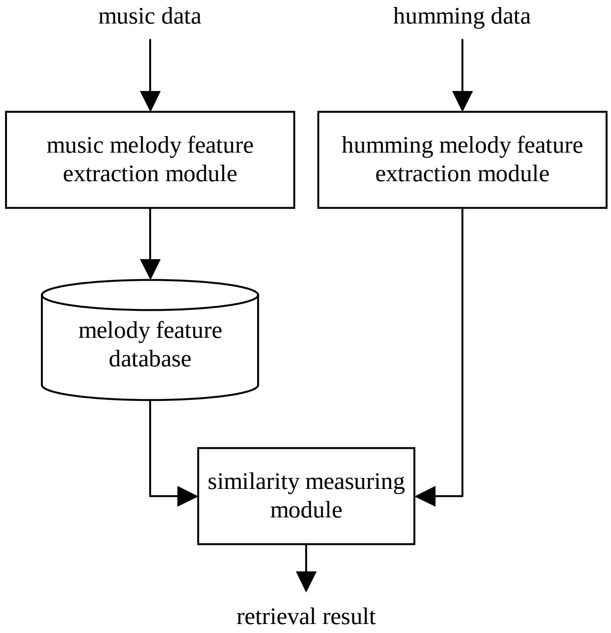
music data
humming data
music melody feature extraction module
humming melody feature extraction module
melody feature database
similarity measuring module
retrieval result

**Figure 1. Overview of QBH system**

humming melody feature extraction module extracts melody feature from humming acquired via stored humming files or microphone. The extracted melody feature is also represented by a sequence of vectors. The music melody extraction module and the humming melody extraction module also contain the process of note segmentation. The note segmentation process groups contiguous audio frames which seem to have a same note. Finally, the similarity measuring module compares music melody feature in the feature database with the humming melody feature and outputs the matching result in the order of their similarity. When measuring the similarity between representations of music melody and humming melody, disparities of overall length and local variances between the music melody in the database and the humming melody are considered.

Additional modules, which are not included in this paper, such as music filtering module that filters out those music with low possibility of being the right answer based on the features other than melody, and melody part detection module which separates and locates melody parts, can be added for completeness of the system and for the better performance in accuracy and speed.

## 3. MELODY FEATURE EXTRACTION

For matching polyphonic audio and humming based on their melody information, we propose to describe such information in a mid-level, in which melody information of music and humming are represented as a set of possible notes of audio frame or audio segment, not as a definite musical note of audio frame or audio segment. Those note candidates of a frame or a segment are selected to be the most audible notes from the sound sources among the notes that are simultaneously generated from multiple sound sources. Audio frames that are believed to have same notes are concatenated to be an audio segment and the note candidates are updated based on the concatenated frames. This process can be divided into five parts as shown in Figure 2.

### 3.1 Harmonic Enhancement

Ordinary music is composed of several sounds from various sound sources, such as vocal, piano, guitar, percussion, and so on, and the source separation and musical note recognition from

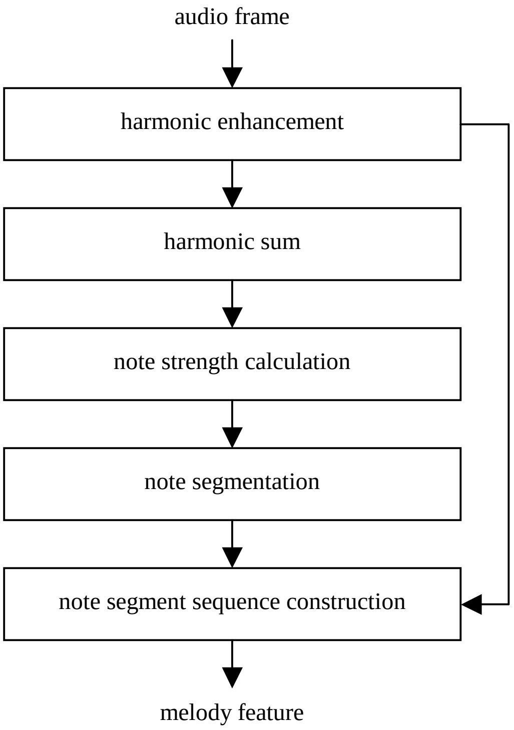

**Figure 2. Melody feature extraction process**

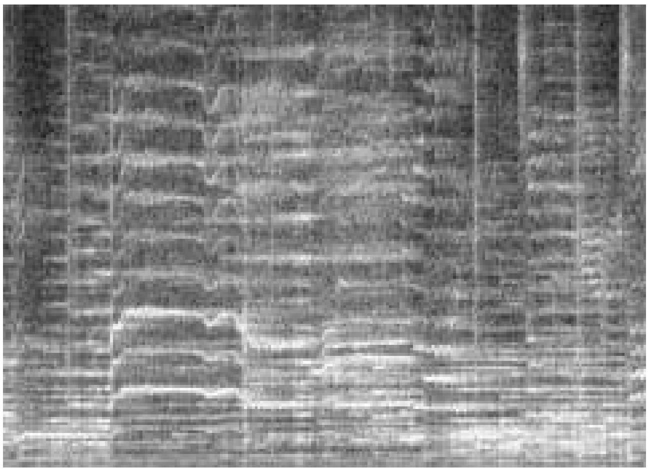

**Figure 3. Spectrogram of music clip**

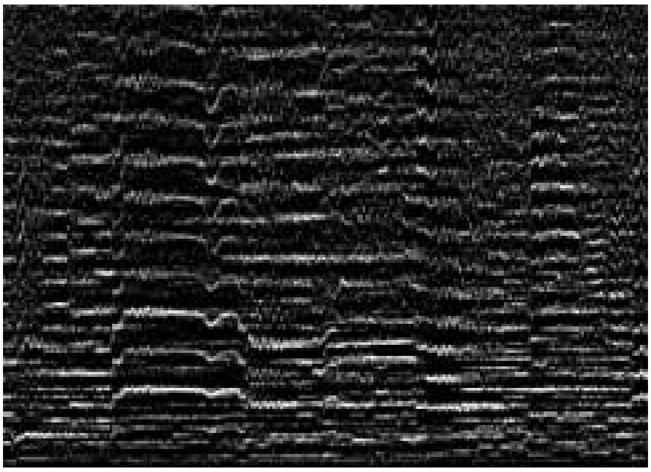

**Figure 4. Enhanced harmonic of music clip**

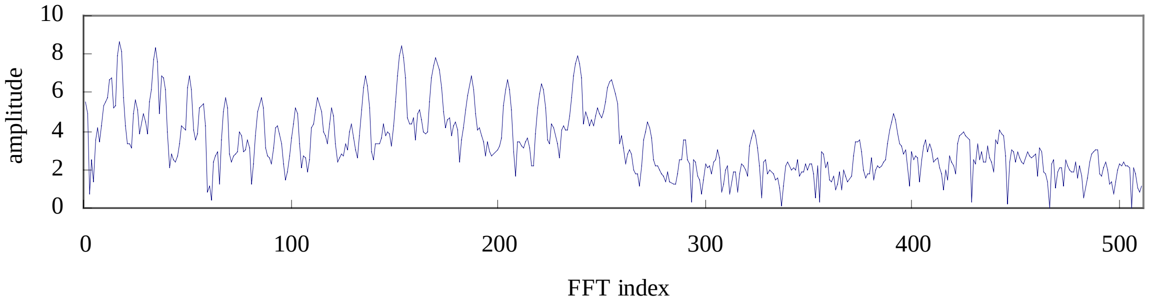

**Figure 5. Spectrogram of music clip at a specific time**

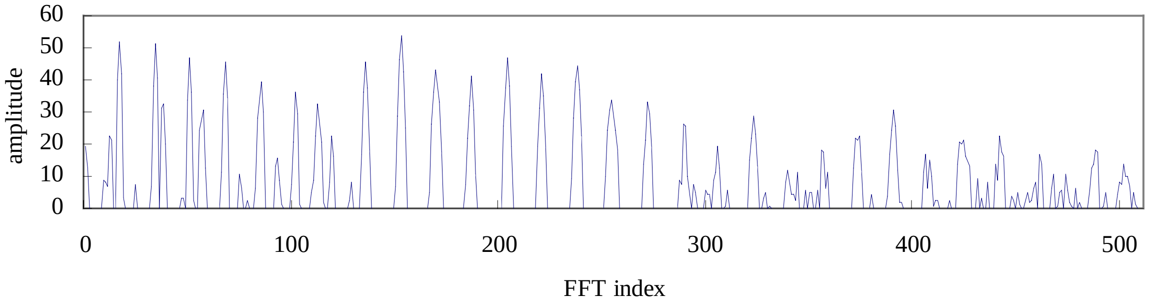

**Figure 6. Enhanced harmonic of music clip at a specific time**

polyphonic raw music are not easy tasks to achieve. But when music retrieval by humming query is the target application, the problem of precise source separation and automatic transcription from polyphonic audio can be avoided, because only the parts and sounds of the music, which are easy for human to recognize and memorize, can be hummed. In other words, only those melodies composed of consecutive sounds that are most audible and predominant than other sounds can be candidates for human to memorize and hum.

Harmonic enhancement process, which is the first step of the melody feature extraction process, extracts those predominant pitches. Musical sounds are constructed by harmonics, which are composed of partials, their positions in frequency and amplitudes, to be specific. Most audible pitch is one that exhibits their clear harmonic structure and has larger amplitude. When the sounds are mixed together, such harmonic structure and amplitude can be impaired by other harmonics. For example, when some harmonics from two sound sources are adjacent in frequency axis, partials of one harmonic can be masked by partials of other harmonics. Consequently, the sound that contains masked partials is less audible than the sound that contains masking partials. In other words, due to the frequency masking effect, a signal with large amplitude tends to perceptually mask other nearby signal in frequency domain. So the sounds that include harmonics of large amplitude have more probability to be the predominant sound. However, even though the amplitudes of harmonics are large, if their surrounding signals also have large amplitudes, those harmonics are less possible to contribute to the predominant sound. For example, if the first harmonic (fundamental frequency) of a sound has large amplitude, but it is superimposed in the band where the noise-like percussion sound is spanning, the predominance of the sound cannot rely on the large amplitude of the first harmonic. Harmonic enhancement process extracts such harmonics that have outstanding peaks compared to the surroundings and the enhanced harmonics can be represented as the equation (1).

$$E_t^{EP}(k) = \sum_{i=-W}^{W} A\big(E_t(k) - E_t(k+i)\big),\ 0 \le k < N \quad \text{- (1)}$$

where $A(x) = x, \forall x \ge 0$ and $A(x) = 0, \forall x < 0$

In equation (1), $N$ is the FFT index range, $E_t^{EP}(k)$ represents the degree of the predominance of the harmonic in the frequency index $k$, considering the spectral amplitudes $E_t(k)$ of surrounding signals within the frequency range $W$ in time $t$. $E_t^{EP}(k)$ is large when the spectral amplitude of the harmonic in the frequency index $k$ is a peak in the spectrum and it becomes much larger when the spectral amplitude of given index $k$ is large compared to its surroundings and when there are no adjacent peaks within the given window size $W$. Figure 3 and 4 shows spectrogram of a music clip and enhanced harmonics respectively. From these two figures, one can easily find that small peaks and wide band signals are eliminated and large and prominent peaks are emphasized. Figure 5 and 6 illustrates this phenomenon more clearly by showing spectrum and enhanced harmonics at a specific time.

## 3.2 Harmonic Sum

The most important ingredient in recognizing musical sound is the harmonicity of the sound. The singing voice of human and the playing sounds of musical instruments show periodic peaks in the frequency axis according to the characteristics of each sound source. It is said that the recognition of the sound is the process of perceiving how much those partials have harmonic characteristics [8]. Pitch extraction using harmonic sum based on above properties has been reported as successful and is adopted as the second step of our melody extraction process. In our algorithm,

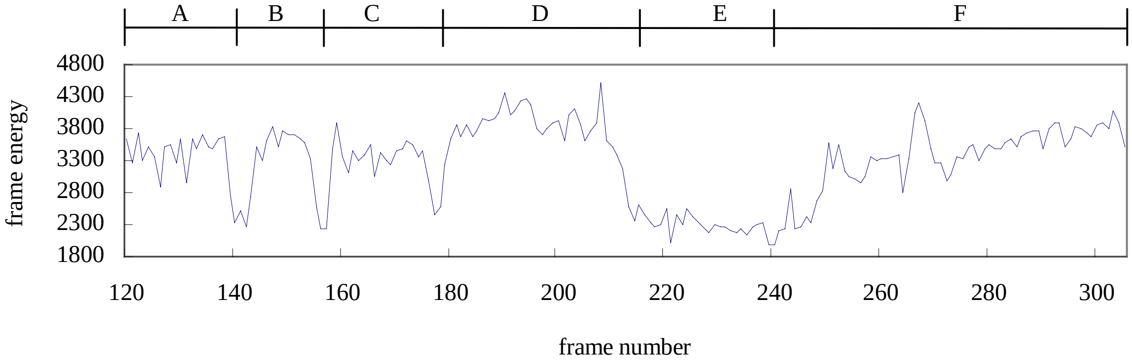

**Figure 7. Frame energy of enhanced harmonics**

the pitch information is extracted using the harmonic sum of the enhanced harmonics as shown in the equation (2).

$$F_t(p) = \frac{1}{\lfloor N/p \rfloor} \sum_{m=1}^{\lfloor N/p \rfloor} E_t^{EP}(mp) \quad \text{- (2)}$$

In equation (2), $\lfloor x \rfloor$ is an integer which is not bigger than $x$ and $F_t(p)$ is the average strength of harmonics having the fundamental frequency $p$ in the audio frame at time $t$. Above equation shows that $F_t(p)$ is decided by averaging the degree of predominance value obtained by harmonic enhancement process in equi-distanced frequency indexes. If there is predominant harmonics of fundamental frequency $p$, the value $F_t(p)$ becomes large and represents the possibility that the sound of fundamental frequency $p$ would occurs is large.

## 3.3 Note Strength Calculation

When a human sings musical notes according to the musical score, it happens that fundamental frequency of the note sung varies slightly, but he/she feels like singing the same note and others cannot distinguish such slight variations. It is also true for playing any musical instrument. Even though there are variations in fundamental frequency and a person can feel them, it is not difficult for people to feel that those variations are within a range of a note and to regard those as same note (like vibrato). In addition to such a psycho-acoustical reason, dimension of the frequency index generated by Fourier transform need to be minimized to speed up the matching by reducing the feature dimension.

Based on above phenomenon and fact, we quantize frequencies into the frequency bands according to the frequencies of musical notes represented in 12-note scale (12 notes per octave). Strength of the fundamental frequency in frequency index acquired by the equation (2) is converted to one in musical note index as shown in equation (3) in the third step.

$$NS_t(m) = \frac{\int_{L_m}^{U_m} F_t(p)\,dp}{|U_m - L_m|}, \quad 0 \le m \le M-1 \quad \text{-(3)}$$

In equation (3), $m$ is a musical note index and $U_m$ and $L_m$ are frequencies of the upper and lower limit for a musical note indexed as $m$, and $M$ is the total number of musical notes in the musical scale. In our approach 512 frequency index is converted to 108 note index (12 notes per octave, 9 octaves). According to the relationship of the note index to frequency [9], frequencies are converted to note index ranging from $C_0$ to $B_8$. For example, $C_0$ corresponds to 16.352Hz and $B_8$ to 7902.1Hz. Frequencies of the upper limit $U_m$ and the lower limit $L_m$ for a note index can be calculated by following equation.

$$\begin{aligned} U_m &= N_m \times \beta \quad \left(= N_{m+1} \times \beta^{-1}\right) \\ L_m &= N_m \times \beta^{-1} \quad \left(= N_{m-1} \times \beta\right) \end{aligned} \quad \text{- (4)}$$

$$N_{m+1} = N_m \times \alpha$$

$$\alpha = 10^{\frac{\log 2}{12}}, \quad \beta = 10^{\frac{\log 2}{24}}$$

In equation (4), $N_m$ is the frequency of note index $m$. $\alpha$ and $\beta$ are the frequency ratio of neighboring note index and the frequency ratio of a note and its boundary, respectively.

## 3.4 Note Segmentation

Consecutive audio frames that show the similar pitch characteristics are grouped into an audio segment in the fourth step. It is known that there is a transient period to reach a specific note when human voices or instrumental sounds occur. The transient period is called onset period, and the sound energy in the period is very low. As we are considering and investigating the prominent sound, we also make use of the measure of prominent sound, that is, the enhanced harmonic data of section 3.1, in the process of note segmentation based on the characteristics of onset period. The segment boundaries are selected as the points that show the local minimum energy of the enhanced harmonics as defined in the equation (5).

$$SB = \{t \mid \min_t(FE(t)), \min(FE(t)) < TH\}$$

$$FE(t) = \frac{1}{N} \sum_{k=0}^{N-1} EP_t^{\,2}(k) \quad \text{- (5)}$$

*FE(t)* is the frame energy of enhanced harmonics at time *t* and *TH* is the threshold value used to avoid selecting too many local minima. Additionally, when the frame energy of the enhanced harmonics shows very small value over a specific duration, that group of frames is classified as silent segment. A segment between two silent segments is merged with the silent segments constructing a single silent segment, if it is shorter than a certain threshold. Figure 7 shows fluctuation of frame energy of enhanced harmonics and segmentation results. Figure shows six segments (A, B, C, D, E, F) produced by the note segmentation process and segment “E” is classified as a silent segment.

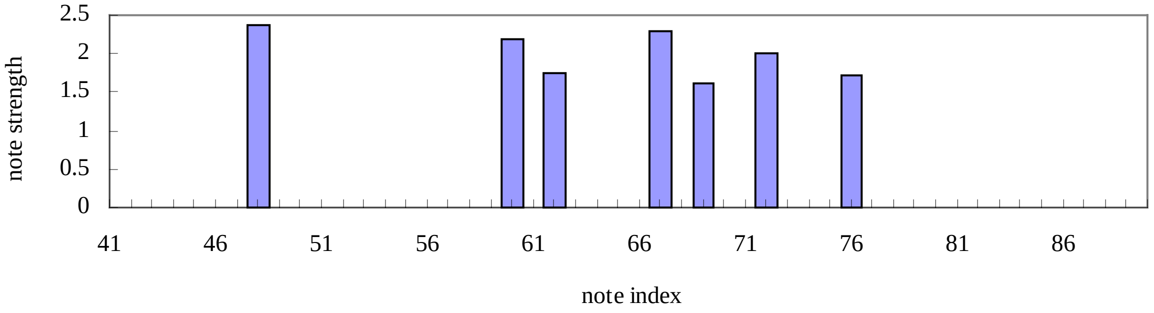

**Figure 8. Note strength of a segment**

## 3.5 Construction of Note Segment Sequence

Each audio segment is composed of several consecutive audio frames and pitch information of the segment is constructed by averaging each component of the note strength vector as shown in equation (6).

$$S_l(m) = \frac{1}{C}\sum_{t=l_s}^{l_e} NS_t(m), \quad 0 \le m \le M-1 \text{ - (6)}$$

In equation (6), $S_l$ is the note strength vector of a segment, $C$ is the number of audio frames included in the segment, and $l_s$ and $l_e$ are the start audio frame and the end audio frame of the note segment respectively, which are obtained by equation (5).

Finally, note candidates are selected by picking several peaks that have large note strength. In our experiments, 7 peaks from the music segment, and 3 peaks from the humming segment are selected. Figure 8 shows the example of the note strength for a segment. Selected peaks are the candidate pitch of a segment and peak values represent the possibility of each peak. The pitch information with their possibility value for consecutive segments constructs the sequence of vectors and it is used as the representation of melody information.

## 3.6 Storing Melody Feature in Feature Database

Extracted melody feature of music is stored in feature database for retrieval process. Melody feature is represented by a sequence of vectors containing note information. Not all the elements of the vector have strength value, because only the note candidates are selected in the process of note segment sequence construction. Only those elements that have note strength information of the note strength vector are saved in database. Hence, a sequence of segment information including pairs of note index and its strength and segment ID describing each segment and a header that contains additional information used in the process of retrieval (number of segments, number of note candidates per segment, music title, etc) are stored for each music clip in the music database, as depicted in figure 9.

| header info. | | | | |
|---|---|---|---|---|
| Seg$_1$ | (NI$_1$, Str$_1$) | (NI$_2$, Str$_2$) | • • • • • | (NI$_L$, Str$_L$) |
| Seg$_2$ | (NI$_1$, Str$_1$) | (NI$_2$, Str$_2$) | • • • • • | (NI$_L$, Str$_L$) |
| | | ⋮ | | |
| Seg$_P$ | (NI$_1$, Str$_1$) | (NI$_2$, Str$_2$) | • • • • • | (NI$_L$, Str$_L$) |

**Figure 9. Data format for storing feature data**

In figure 9, Seg$_P$ stands for segment ID of *p*th segment, and NI$_x$, and Str$_x$ means the note index and its strength of *x*th note candidate of the given segment, respectively. L and P in the figure means the number of note candidates in a segment and the number of segments in the music clip respectively.

# 4. MATCHING

Humming query can have disparities with music in their length. Also, erroneous notes can be inserted and some notes can be even omitted. Furthermore, erroneous note and segment information can also be extracted during melody feature extraction process.

To overcome such erroneous environment, DP matching method is used to match two patterns of different length while permitting partial variations and errors [10]. Because there also exist overall biases in pitch between music and humming, DP array is generated using note strength vector while shifting the vector index.

## 4.1 Dissimilarity Calculation using DP Matching

Let sequences of music segments and humming segments, which are constructed by the vector containing the information of note,

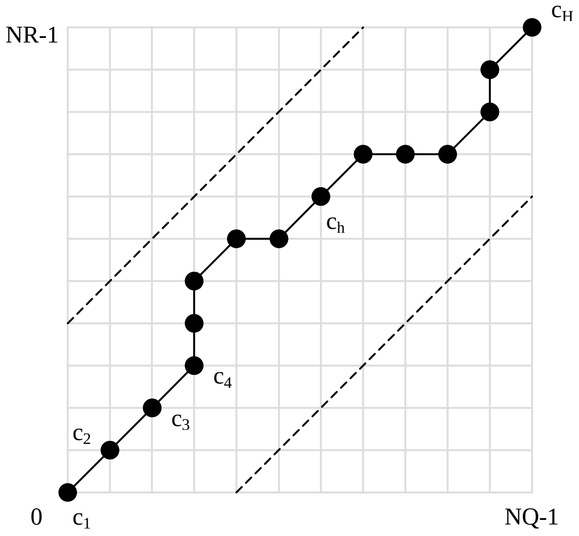

**Figure 10. Matching path on DP array**

be *R* and *Q*,

$$R = [r_0, r_1, r_2, \ldots, r_i, \ldots, r_{NR-1}]$$
$$Q = [q_0, q_1, q_2, \ldots, q_j, \ldots, q_{NQ-1}]$$

where $r_i$ is the note strength vector of *i*th segment of music and $q_j$ is the note strength vector of *j*th segment of humming. *NR* and *NQ* are the number of segments in the music and humming respectively.

To match two sequences *R* and *Q*, a matrix *D* of size *NR* × *NQ* is constructed. An element of the matrix, $d_{ps}(r_i, q_i)$ as given in equation (7), represents the dissimilarity between $r_i$ and $q_j$ when the overall pitch shift is *ps*.

$$d_{ps}(r_i, q_j) = \frac{\sqrt{\sum_{m=0}^{M-1}\left[r_i(m) - q_j(m-ps)\right]^2}}{\sqrt{\sum_{m=0}^{M-1} r_i^{\,2}(m)}\sqrt{\sum_{m=0}^{M-1} q_j^{\,2}(m-ps)}}, 0 \le m, m-ps \le M-1 \text{- (7)}$$

The matching path $C_{ps}$ in case of pitch shift *ps* is defined as a set of consecutive vector elements $d_{ps}(r_i, q_i)$ which decides the matching between *R* and *Q* (figure 10). The *h*th element of matching path $C_{ps}$ is defined as $c_{ps,h}=(i, j)$ and the matching path can be represented as the following equation assuming that the length of the matching path is *H*.

$$C_{ps} = c_{ps,1}, c_{ps,2}, c_{ps,3}, \ldots, c_{ps,h}, \ldots c_{ps,H} \quad \text{- (8)}$$
$$\max(NR, NQ) \le H < NR + NQ + 1$$

After generating DP arrays for various pitch shift values, matching path can be selected for the matching cost to be minimized for each pitch shift values.

$$DP_{ps}(C_{ps,\min}) = MIN\left\{\frac{\sqrt{\sum_{h=1}^{H_{ps}} d_{ps}\left(c_{ps,h}\right)}}{H_{ps}}\right\} \text{- (9)}$$

Final dissimilarity value between *R* and *Q* is calculated by selecting matching cost of a pitch shift that has minimal matching cost among several pitch shift values as shown in Equation (9). In equation (9), $DP_{ps}(C_{ps,\min})$ is the selected matching cost and $H_{ps}$ is the length of matching path with pitch shift *ps*.

## 4.2 Windowing

To reduce the matching time by avoiding invalid matching path that is too far from the ideal diagonal matching path, we used windowing method on DP arrays. It is depicted in figure 10 as dotted lines. When the width between upper and lower dotted line is narrow, the matching time is reduced, but the degree of allowed variations between music and humming is lowered, hence the matching becomes even strict.

## 4.3 Additions to the Conventional DP Matching

We added the measure that reflects the amount of how far the matching path is from the diagonal path, and it give more emphasis on matching paths generated along with the diagonal line in DP array. This is calculated by the following equation.

$$f_{ps}(R,Q) = {H_{ps}}\Big/{NR + NQ}$$

where $H_{ps}$ is the length of the matching path.

This measure is applied to the matching cost $DP_{ps}$ as a normalization factor for calculating dissimilarity value between *R* and *Q* in the form of addition or multiplication.

# 5. EXPERIMENTATION

## 5.1 Experiment Configuration

Experiment was executed at the music database that is composed of 92 melody clips. Each melody clip contains melody part of music and is about 15~20 seconds long. These clips are selected from Korean and Western popular songs. 176 humming samples from 7 males and 3 females are gathered using microphone.

All the data is stored in PCM format with mono, 16kHz sampling rate and 8 bits/sample resolution. The experiments are performed on a system with Windows 2000 O/S and Pentium IV 1.5GHz CPU.

## 5.2 Experimental Results

Experiment was done according to the various conditions, such as the size of neighbors in partial enhancing process (W=4, W=8) and usage of *f* measure which measures how far the matching path is from the diagonal path (NNF: no use, NNF1: multiplication, NNF2: addition) as the following table.

**Table 1. Configuration of experiment**

| | NNF | NF1 | NF2 |
|---|---|---|---|
| **W=8** | M01 | M02 | M03 |
| **W=4** | M04 | M05 | M06 |

Table 2 shows the retrieval performance. Top *n* means the rate of queries that retrieves correct music within top *n* rank. For example, when the experiment configuration is M03 (W=8, use of *f* measure with addition to DP distance value), we can say that the number of cases of retrieving correct music as top 1 is more than 4 times from 10 trial and the number of cases of retrieving within top 10 is over 7 times. In general, result shows that when the size of neighboring frequency index to be considered (*W*) in the process of harmonic enhancing is 4, the retrieval accuracy is better than the case of *W*=8, and the usage of measure that measures how far the matching path is from the diagonal path (i.e., *f* measure) is helpful.

**Table 2. Retrieval accuracy according to various experimental configurations defined in table 1.**

| | Top 1 | Top 3 | Top 5 | Top 10 |
|---|---|---|---|---|
| **M01** | 11.93 | 23.30 | 34.66 | 50.56 |
| **M02** | 43.75 | 52.84 | 57.95 | 67.05 |
| **M03** | 43.18 | 56.82 | 64.77 | 76.14 |
| **M04** | 30.11 | 49.43 | 60.23 | 68.75 |
| **M05** | 40.91 | 60.80 | 67.05 | 76.13 |
| **M06** | 37.50 | 58.52 | 65.91 | 75.57 |

When windowing on DP arrays is used, the matching time was greatly reduced and still preserves the retrieval rate. The size of window was decided proportional to the length of music clip and humming. Following is the retrieval rate and the matching time according to the various window sizes. It shows that the matching time is reduced while the retrieval rate is not noticeably impaired.

**Table 3. Experimental result of various window size on matching path and corresponding matching time**

| M3 | Matching time (sec) | Top 1 | Top 3 | Top 5 | Top 10 |
|---|---|---|---|---|---|
| **W0** | 14.15 | 43.18 | 56.82 | 64.77 | 76.14 |
| **W1** | 10.64 | 43.18 | 56.82 | 64.77 | 76.14 |
| **W2** | 6.95 | 43.18 | 56.82 | 64.77 | 76.14 |
| **W3** | 4.42 | 43.75 | 56.82 | 65.91 | 75.00 |
| **W4** | 2.89 | 42.61 | 54.55 | 61.93 | 69.31 |

W0, W1, W2, W3 and W4 are specification of size of window such as no windowing, (NR+NQ)/2, (NR+NQ)/4, (NR+NQ)/8, and (NR+NQ)/16, respectively.

## 6. CONCLUSION

This paper presented the method of matching humming query with polyphonic music data. Focus on this paper is to extract melody information from polyphonic music and to match the melody information in the mid-level representation. For this purpose, the melody feature extraction module contains the process of harmonic enhancement that emphasizes the most audible sound from sound mixture, the harmonic sum that calculates pitch candidates and their possibility of occurrence, the note strength calculation that converts amplitudes in frequency index to those in musical note index, the note segmentation that groups adjacent frames and extracts note duration according to the frame energy of the enhanced harmonics, and the note segment sequence construction that extracts note information of the segment and constructs melody feature. Melody feature extracted from music clip have couples of note candidates and feature from humming may have erroneous notes that are slightly different from right answer. To overcome this fact, DP matching method is adopted. Experimental results show the possibility of matching humming with music in the mid-level representation within proper matching time. We can avoid exact transcription problem by representing melody information as a sequence of vectors that contain the information of pitch and its possibility of occurrence. The method proposed in this paper is only performed on a clip-based database and DP matching of a humming with a complete clip is performed. To overcome the given constraint in a large database of complete songs, additional processes may be adopted. First, beat/tempo analysis with database indexing such as R*-tree would make it possible to filter out many songs from the candidates so as to reduce matching time. Second, detection of important melody parts, such as beginning melody part and repetitive part of a song, and approximate subsequence matching of the detected melody parts can be applied to matching of longer songs. Further researches will be conducted to perform matches on a database that contains features extracted from complete songs based on the technique proposed in this paper.

## 7. REFERENCES

[1] H.Y. Tseng, "Content-based Retrieval for Music Collections," Proc. of 4th ACM conference on Digital Libraries, pp.176~182, California, 1999.

[2] Y. Kim, W. Chai, R. Garcia, B. Vercoe, "Analysis of a Contour-Based Representation for Melody," Proc. International Symposium on Music Information Retrieval, Oct. 2000.

[3] C.-C. Liu, J.-L. Hsu, A.L.P. Chen, "An Approximate String Matching Algorithm for Content-based Music Data Retrieval," IEEE Int. Conf. on Multimedia Computing and Systems, vol.1, pp.451~456, 1999.

[4] R.J. McNab, L. Smith, I.H. Witten, C.L. Henderson, "Tune Retrieval in the Multimedia Library," Multimedia Tools and Applications, vol.10, pp.113~132, 2000.

[5] A. Klapuri, "Multipitch Estimation and Sound Separation by the Spectral Smoothness Principle," IEEE International Conference on Acoustics, Speech and Signal Processing, vol.5, pp.3381~3384, 2001.

[6] T. Miwa, Y. Tadokoro and T. Saito, "Musical Pitch Estimation and Discrimination of Musical Instruments Using Comb Filters for Transcription," 42nd Midwest Symposium on Circuits and Systems, vol.1, pp.105~108, 2000.

[7] M. Goto, "A Predominant-F0 Estimation Method for CD Recordings: MAP Estimation Using EM Algorithm for Adaptive Tone Models," IEEE International Conference on Acoustics, Speech and Signal Processing, vol.5, pp.3365~3368, 2001.

[8] A. Klapuri, "Pitch Estimation Using Multiple Independent Time-Frequency Windows," IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pp.115~118, 1999.

[9] Thomas D. Rossing, *The Science of Sound*, Addison-Wesley Publishing Company, 1990.

[10] L.Rabiner, B. –H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.