An Effective Music Information Retrieval Method Using Three-Dimensional Continuous DP

Sung-Phil Heo, Motoyuki Suzuki, Akinori Ito, and Shozo Makino

***Abstract*—This paper describes a music information retrieval system that uses humming as the key for retrieval. Humming is an easy way for a user to input a melody. However, there are several problems with humming that degrade the retrieval of information. One problem is the human factor. Sometimes, people do not sing accurately, especially if they are inexperienced or unaccompanied. Another problem arises from signal processing. Therefore, a music information retrieval method should be sufficiently robust to surmount various humming errors and signal processing problems. A retrieval system has to extract the pitch from the user's humming. However, pitch extraction is not perfect. It often captures half or double pitches, which are harmonic frequencies of the true pitch, even if the extraction algorithms take the continuity of the pitch into account. Considering these problems, we propose a system that takes multiple pitch candidates into account. In addition to the frequencies of the pitch candidates, the confidence measures obtained from their powers are taken into consideration as well. We also propose the use of an algorithm with three dimensions that is an extension of the conventional Dynamic Programming (DP) algorithm, so that multiple pitch candidates can be treated. Moreover, in the proposed algorithm, DP paths are changed dynamically to take deltaPitches and IOIratios (inter-onset-interval) of input and reference notes into account in order to treat notes being split or unified. We carried out an evaluation experiment to compare the proposed system with a conventional system [6]. When using three-pitch candidates with conference measure and IOI features, the top-ten retrieval accuracy was 94.1%. Thus, the proposed method gave a better retrieval performance than the conventional system.**

***Index Terms*—Continuous DP, dynamic melody representation, humming, multiple pitch candidates, music information retrieval.**

I. Introduction

Multimedia information technologies, which provide comprehensive and intuitive information for a broad range of applications, have a strong impact on modern life and have changed our thinking and their usage. Over the past few decades, there has been an explosive growth in the use of digital multimedia (including audio, music, video and images) over the Internet and wireless communications.

In order to make effective use of the huge amount of multimedia contents, information retrieval systems are needed. Conventional information retrieval (IR) research has been based mainly on text information [1]–[3]. However, this research cannot be directly applied to a multimedia information retrieval system because the text-based IR method is to retrieve desired text documents using a search query consisting of a number of text-based keywords. For example, in conventional music information retrieval (MIR) systems, retrieval keys consist mainly of text information such as a singer's name, a composer, the title of a piece of music, or the lyrics of a song [1], [3]. On the other hand, several recent MIR systems utilize music information extracted from humming as the key for information retrieval.

The full query-by-humming system was first proposed by Ghias *et al.* [5], and several such systems have been developed, including MELDEX [14], Themefinder [15], TuneServer [16], MiDiLiB [17], Super MBox [18], SoundCompass [19], and others. These systems use various pitch extraction, melody representation and matching methods.

Manuscript received April 1, 2004; revised May 30, 2005. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Radu Serban Jasinschi.

S. Heo is with the Service Development Laboratory, Korea Telecom, Seoul 137-792, Korea.

M. Suzuki, A. Ito, and S. Makino are with the Graduate School of Engineering, Tohoku University, Sendai 980-8579, Japan.

Digital Object Identifier 10.1109/TMM.2006.870717

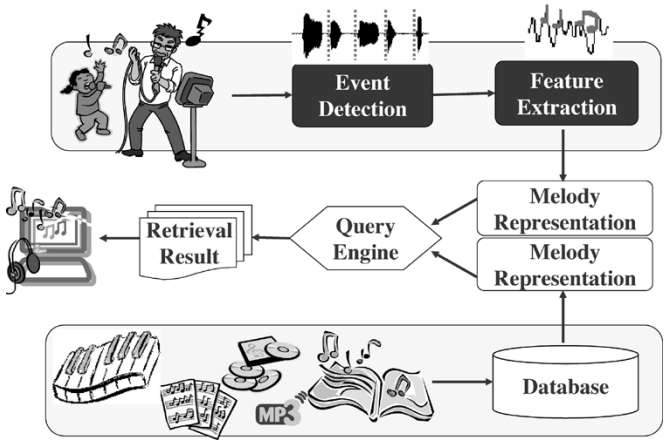

Fig. 1. Overview of music information retrieval system.

An overview of an MIR system using query-by-humming is shown in Fig. 1. There are several main components in this system: an event detection module, a feature extraction module, a melody representation module and a similarity measurement module. First, a user hums a melody to input the music information. The purpose of event detection is to identify each note's onset and/or offset boundaries within the acoustic signal. The system detects notes from the humming and then extracts features, such as durations (sound lengths) and pitches. The similarity measurement engine works by using the extracted features between the humming and the database, and the query engine carries out the matching of the humming to the database, and the nearest matching melodies found in the database become the results. A ranked list of matching melodies is then displayed on the screen.

The techniques applied in an MIR system using query-by-humming should be effective, i.e. it should return precise results. In order to develop an effective MIR, several issues should be considered.

- Effective melody representation and matching method
  First, the tonality and tempo of a humming might differ from user to user. Almost all MIR systems employ relative-value based melody representation to absorb the differences between the hummed melody and melodies in the database. Relative pitch, which is called *deltaPitch*, is calculated by the subtraction of the successive note from the pitch. Relative Inter Onset Interval (IOI) ratio, which is called the *IOIratio*, is calculated by dividing the successive note by the IOI.
  QBH [5] and TuneServer employ symbol-based melody representation. This representation expresses melody sequence as a string consisting of three kinds of symbols ("U," "D," *and* "S," which signify that a note is higher than, lower than, or the same as the previous note, respectively). This melodic representation (coarse melodic contour) can absorb the instability of a user's pitch. However, the problem of this method is its retrieval performance. Because of the coarse melody contour, one humming often matches more than one piece of music. Thus, the melody representation has to be more precise (thus, less compact) in order to ensure better discrimination. Several MIR systems such as MELDEX and MiDiLiB employ the improved version of the coarse melodic contour in melody representation. On the other hand, Utagoe [6] proposed a *dynamic threshold determination* method for melody representation. This method can obtain the approximate relative-value sequence of pitch/IOI with the maximum quantity of information by determining thresholds dynamically. Another approach is the *beat-based* method. It is proposed in SoundCompass, but this method gives a constraint of tempo using a metronome.

Next, singing errors, such as the insertion or omission of notes, are contained in the user's humming because of imperfect recall. Sometimes, event detection errors such as notes being split or unified occur when segmenting the input humming into individual notes. Therefore, a note in a humming might be split into two notes, or united with a successive note. Most MIR systems use a dynamic programming (DP) matching algorithm [4], [6], [22], [23]. A DP matching algorithm can consider the insertion or omission of notes, and it gives the optimum correspondence between the humming and the database.

- Accurate pitch extraction and matching method
  It is difficult to extract the true pitch perfectly from the humming. Conventional MIR systems use high-precision pitch extraction algorithms. However, the pitch extraction accuracy by those algorithms is not perfect.
  QBH tracks pitch using the autocorrelation method, converts it to coarse melodic contours, and matches the contour against a database. The retrieval is carried out based on an approximate string-matching algorithm that allows the transposition, dropout and duplication of notes.
  Utagoe uses frequency-based pitch extraction and proposed the *coarse-to-fine matching*, effectively reducing the number of answer candidates by considering the trade-off between coarse matching and fine matching. Coarse matching can tolerate input errors but it cannot reduce the number of answering candidates, while fine matching does the opposite. However, pitch extraction errors may occur when extracting the features.
  Most of the errors obtained in pitch extraction were caused by capturing a double or a half pitch. The narrower the pitch range, the easier the problem is to solve. However, in music, pitch can have leaps of one octave or more. Octave errors are the most common error observed in MIR systems. Therefore, if multiple pitches are used as candidates in consideration of errors of pitch extraction, the harmonic frequency detection problem can be solved.

Although conventional MIR systems propose error-tolerant melody representation, pitch extraction and matching method, they will still have the following problems. First, when notes are split or unified by singing error and/or event detection error, relative-values are changed since a successive note is changed. Next, even if a hummed query is perfect, it is still difficult to extract the true pitch perfectly from the hummed query. Since the true pitch extraction problem generated at the time of signal processing remains as usual even if a user's humming is perfect.

We believe that the above problems are the key points for the realization of a robust and efficient MIR system [12]. To deal with these problems, we need an effective melody representation method, a musically reasonable matching method, and a true pitch extraction method. In this paper, we detail conventional problems, and propose a new error-tolerant MIR method.

II. Melody Representation and DP Matching

When a user inputs a humming, the following things should be considered: 1) even if a user hums the same notes, there are variations in tempo and tonality; 2) splitting and/or unification of notes occur due to the ambiguous memory of a user or the interval detection errors in the MIR system. Here, considering (1), the variation is absorbed by taking relative-values between successive notes. With respect to (2), the variation can be absorbed by using DP matching [11], [13]. However, DP matching cannot treat the splitting or unification of notes as long as a simple relative-value-based melody representation is employed. We refer to the ordinary relative-value-based melody representation as *static melody representation*. In this section, we first explain the conventional DP matching method and the problem of static melody representation. Then we propose a dynamic melody representation.

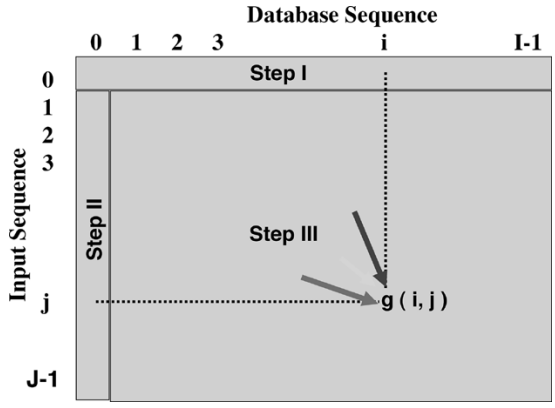

Fig. 2. Conventional DP algorithm.

A. Static Melody Representation in Conventional DP

Here, we first explain the conventional continuous DP matching [5], [6], [22], [23]. The continuous DP matching is a method to match an input sequence to a part of a reference sequence. Let the length of the input note sequence be J, and that of the reference note sequence in the database be I. Then, the DP distance between a part of the reference note sequence and the input note sequence can be calculated using the DP matrix $g(i,j)$, such as that illustrated in Fig. 2. The pitch values are expressed by *cent-based log frequency*. This is calculated according to $d(f) = 1200(log_2 f - log_2 f_0)$. Here, f_0 and f stand for the reference frequency and the concerned frequency, respectively. One cent is equal to 1/100 of a semitone.

First, the distance between two notes is defined. Let $m_p(i)$ be the cent-based log frequency of the ith note in the reference sequence, and $h_p(j)$ be the cent-based log frequency of the jth note in the input sequence. Let us consider the deltaPitches of the two successive notes as

$$\Delta m_p(i) = m_p(i+1) - m_p(i) \tag{1}$$

$$\Delta h_p(j) = h_p(j+1) - h_p(j). \tag{2}$$

Then, the distance of the deltaPitch $p(i,j)$ is defined as

$$p(i,j) = |\Delta m_p(i) - \Delta h_p(j)|. \tag{3}$$

Similarly, the distance of the IOI difference $t(i,j)$ is defined as

$$t(i,j) = |\Delta m_t(i) - \Delta h_t(j)| \tag{4}$$

where $m_t(i)$ is the IOI of the ith note in the reference sequence and $h_t(j)$ is the IOI of the jth note in the input. $\Delta m_t(i)$ and $\Delta h_t(j)$ are defined as follows:

$$\Delta m_t(i) = \log \frac{m_t(i+1)}{m_t(i)} \tag{5}$$

$$\Delta h_t(j) = \log \frac{h_t(j+1)}{h_t(j)}. \tag{6}$$

The distance between the input note and the reference note is

$$d(i,j) = \gamma p(i,j) + (1-\gamma)t(i,j) \tag{7}$$

where γ is the weighting factor.

Next, the DP distance between the input and reference sequences is calculated as follows:

$$g(i,-1) = \infty \quad (0 \le i \le I-1) \tag{8}$$
$$g(-1,j) = \infty \quad (0 \le j \le J-1) \tag{9}$$
$$g(i,0) = d(i,0) \quad (0 \le i \le I-1) \tag{10}$$
$$g(0,j) = \infty \quad (1 \le j \le J-1) \tag{11}$$
$$g(i,j) = \min \begin{cases} g(i-2,j-1) + d(i,j) \\ g(i-1,j-1) + d(i,j) \\ g(i-1,j-2) + 2d(i,j) \end{cases} \quad (1 \le i \le I-1, 1 \le j \le J-1). \tag{12}$$

Here, $g(i,j)$ can be calculated step by step by increasing i and j. Then, $g(i,J-1)$ gives the shortest DP distance between the input note sequence and any subsequence of the reference sequence that ends at the position i. Finally, the shortest distance between the input sequence and any subsequence in the reference sequence is given by

$$D = \min_i g(i,J-1). \tag{13}$$

As explained above, the conventional DP matching depends on the normalization of the tonality and the tempo on the difference-based melody representation, $\Delta m_p(i)$, $\Delta h_p(j)$, $\Delta m_t(i)$, and $\Delta h_t(j)$. The conventional melody representation methods normalize the features by calculating pitch and IOI values relative to the successive notes [5], [6], [22], [23].

However, there are several problems with the conventional melody representation method. As a relative-value of a note is calculated using the pitch value and the IOI of the successive note, the relative-value changes when the notes are split or unified. Although both deltaPitch and IOIratio representation are problems when notes are split or unified, the problem of deltaPitch representation is explained first. Fig. 3 shows an example of the effect of splitting a note. In this figure, a note G in the humming is recognized as two Gs. The deltaPitch sequence in the database is {200 cents, 200 cents, 300 cents}, while the sequence obtained from the humming is {200 cents, 200 cents, 300 cents, 0 cents}. The distance between these sequences is {300 cents}. This mismatch arises from the fact that the calculation of the deltaPitch does not take the possibility of splitting or union of notes into account. Therefore, we propose a new melody representation strategy to treat this type of mismatch. To this end, (3) must be modified with the DP paths because the deltaPitch values should be dynamically calculated considering the splitting or unification of notes.

B. Dynamic Melody Representation

When a note is matched based on the hypothesis that the previous note in the humming is split into two notes, the deltaPitch of the current note in the humming should be considered in relation to the second note before the current note. Then, the deltaPitch of the last note in the humming data will become $\{G - E = 300 \text{ cents}\}$ instead of $\{G - G = 0 \text{ cent}\}$.

This means that the relative-value must be dynamically determined. In the case of unification, the deltaPitches in the database sequence must be dynamically determined by the same method as well.

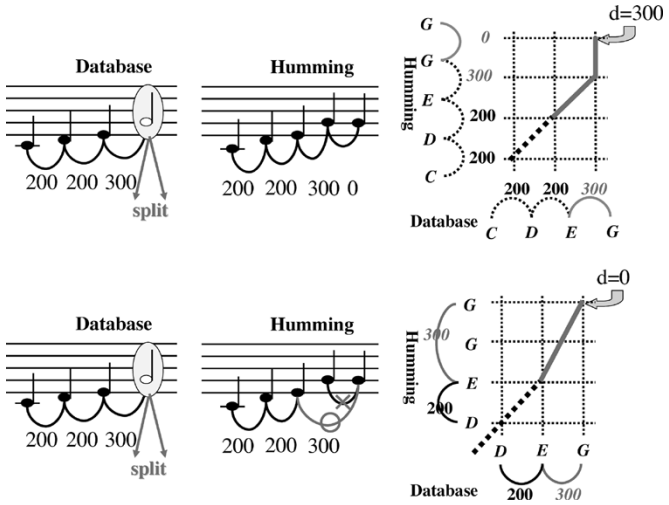

Fig. 3. DeltaPitch value conversion method when splitting occurs. (Upper: conventional method; lower: proposed method).

This problem can be solved by considering two deltaPitches for one note:

$$\Delta_1 m_p(i) = m_p(i+1) - m_p(i) \tag{14}$$
$$\Delta_2 m_p(i) = m_p(i+1) - m_p(i-1) \tag{15}$$
$$\Delta_1 h_p(j) = h_p(j+1) - h_p(j) \tag{16}$$
$$\Delta_2 h_p(j) = h_p(j+1) - h_p(j-1). \tag{17}$$

Δ_1 values are the same as that of the conventional DP in (1) and (2). Δ_2 values are differences between the current note and the second note before the current note. Then, we consider three kinds of distances between two deltaPitches, p_1, p_2 and p_3. These distances correspond to the union case, the no splitting nor unification case, and the splitting case, respectively:

$$p_1(i,j) = |\Delta_2 m_p(i) - \Delta_1 h_p(j)| \tag{18}$$
$$p_2(i,j) = |\Delta_1 m_p(i) - \Delta_1 h_p(j)| \tag{19}$$
$$p_3(i,j) = |\Delta_1 m_p(i) - \Delta_2 h_p(j)|. \tag{20}$$

In the case of the IOI, we need another approach. Fig. 4 shows a case where note $m3$ and $m4$ in the reference sequence are split or united in the humming sequences. When $m3$ and $m4$ are hummed as one note (HUM sequence A), the IOIratio between $a2$ and $a3$ must be compared with that between $m2$ and $m3 + m4$. In the same manner, when $m3$ is hummed as two notes (HUM sequence B), the IOIratio between $b2$ and $b3' + b3''$ corresponds to that between $m2$ and $m3$.

Therefore, we use two IOIratios for one note, as follows:

$$\Delta_1 m_t(i) = \log \frac{m_t(i+1)}{m_t(i)} \tag{21}$$
$$\Delta_2 m_t(i) = \log \frac{m_t(i) + m_t(i+1)}{m_t(i-1)} \tag{22}$$
$$\Delta_1 h_t(j) = \log \frac{h_t(j+1)}{h_t(j)} \tag{23}$$
$$\Delta_2 h_t(j) = \log \frac{h_t(j) + h_t(j+1)}{h_t(j-1)}. \tag{24}$$

Then, we consider three kinds of distances between two IOIratios, t_1, t_2 and t_3, just like in the distances of deltaPitches.

$$t_1(i,j) = |\Delta_2 m_t(i) - \Delta_1 h_t(j)| \tag{25}$$
$$t_2(i,j) = |\Delta_1 m_t(i) - \Delta_1 h_t(j)| \tag{26}$$
$$t_3(i,j) = |\Delta_1 m_t(i) - \Delta_2 h_t(j)|. \tag{27}$$

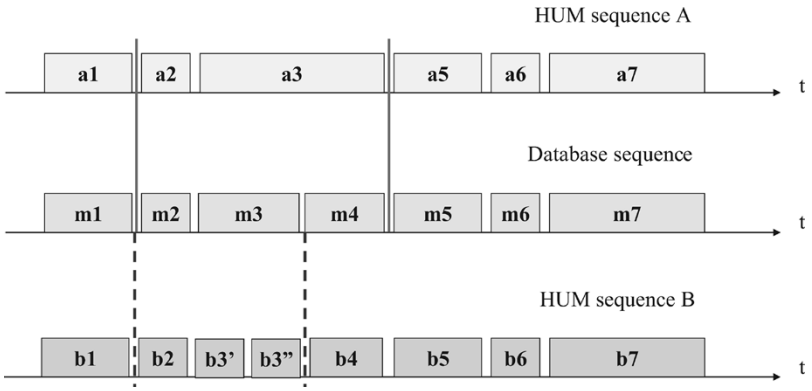

Fig. 4. IOIratio value conversion method when splitting or union occurs.

Finally, the accumulation distance is modified as follows:

$$g(i,-1)=\infty \quad (0\le i\le I-1) \tag{28}$$

$$g(-1,j)=\infty \quad (0\le j\le J-1) \tag{29}$$

$$g(i,0)=d_2(i,0) \quad (0\le i\le I-1) \tag{30}$$

$$g(0,j)=\infty \quad (1\le j\le J-1). \tag{31}$$

$$g(i,j)=\min\begin{cases} g(i-2,j-1)+d_1(i,j) \\ g(i-1,j-1)+d_2(i,j) \\ g(i-1,j-2)+2d_3(i,j) \end{cases} \quad (1\le i\le I-1, 1\le j\le J-1) \tag{32}$$

$$d_e(i,j)=\gamma p_e(i,j)+(1-\gamma)t_e(i,j) \quad (e=1,2,3). \tag{33}$$

In ordinary DP matching, the value at a certain point $d(i,j)$ is independent of the DP paths and only depends on i and j. In the proposed (33), however, the values at the same point $d_{e=1,2,3}(i,j)$ are changed according to the DP paths. When calculating the similarity, this equation can dynamically change the notes that correspond to the DP paths.

III. Feature Extraction

Pitch information is the most important feature for humming-based music information retrieval systems. Therefore, the accuracy of pitch extraction greatly affects the system's performance [8], [24], [25]. Generally, an MIR system extracts one pitch from one note and uses it as a feature value of the note. Recently, several pitch extraction algorithms [24], [25] have been developed that give high accuracy results. However, even these algorithms do not have a 100% accuracy because the harmonic frequency extracted in a music signal contains a higher frequency band than a voice signal. Therefore, we consider multiple pitch candidates to enhance the performance of retrieval accuracy. In this section, a method is described that calculates the multiple pitch candidates as well as their confidence measures.

Multiple pitch extraction is based on cepstral analysis [7]. Fig. 5 shows the basic flowchart of multiple pitch candidates and confidence measures. First, the power spectrum is obtained from the input signal using FFT. Next, the logarithm and IFFT are applied to obtain the cepstrum. Then, the cepstral peaks in the fundamental frequency's range of existence are chosen as pitch candidates. Finally, the quefrencies of the peaks are converted into pitch frequencies. Multiple pitch candidates (MPC) are passed to the query engine without choosing one candidate in the feature extraction module.

Next, confidence measures (CM) are calculated from the values of cepstral peaks. The confidence measures are calculated as the cepstral value of the peak divided by that of the top candidate.

We investigate the accuracy of the pitch extraction when multiple pitch candidates are considered. The accuracy of the pitch extraction by YIN is also shown for comparison. YIN outputs only one pitch as a extraction result.

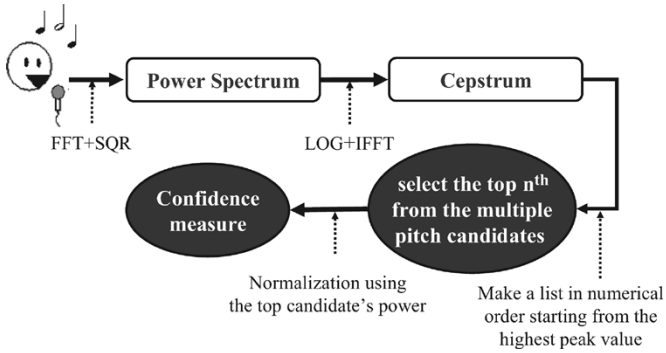

Fig. 5. Extraction flow of multiple pitch candidates and confidence measure.

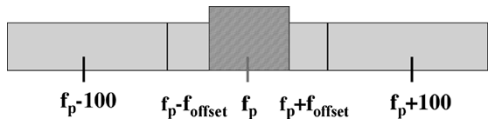

Fig. 6. Range of pitch extraction accuracy.

TABLE I
Evaluation of Pitch Extraction Accuracy

Ranks	Extraction accuracy
The 1st rank	88.4%
Within 2 ranks	96.3%
Within 3 ranks	99.7%
YIN	96.1%

The pitch extraction accuracy was calculated, as shown in Fig. 6, by comparing the correct pitch frequency with the extraction result. An extracted pitch value is regarded to be correct when the difference between the log frequency of the correct pitch and that of the extracted pitch is less than 20 cents. f_p is a cent-based log frequency of the correct pitch, i.e.

$$f_p = 1200\log_2 f \tag{34}$$

where f stands for the correct pitch in Hz. The absolute value of f_p is represented so that the 1 Hz signal corresponds to 0. It does not have any musical meaning, but the difference of two cent-based log frequencies is equivalent to the fraction of two pitch frequencies represented by a cent. A cent-based log frequency of the detected pitch f_d is regarded to be correct when

$$|f_d - f_p| \le f_{offset} \tag{35}$$

where $f_{offset} = 20$ cents.

The references were manually labeled and the accuracy of pitch extraction was analyzed using 260 pieces of data hummed by five subjects. Table I shows the pitch extraction accuracy of MPC and YIN.

The pitch extraction accuracy of the cepstrum-based method was lower than that of YIN, but the accuracy became sufficiently high when multiple pitch candidates were considered. Most of the errors obtained by MPC and YIN were caused by capturing the double or half pitch.

Harmonic frequencies (double or half pitch of the true pitch) were extracted at most frames with incorrect results.

By using three pitch candidates, the accuracy became 99.7%. This result shows that three pitch candidates are sufficient for the subsequent processing.

IV. Three-Dimensional Continuous DP Algorithm

The features obtained from humming and the features of musical pieces in the database are matched using continuous DP. However, the matching algorithm must be extended so that multiple pitch candidates can be utilized along with confidence measures. In this section, the DP algorithm explained in Section II is modified into the three-dimensional (3-D) DP algorithm so that the proposed features can be treated.

First, let $h_p(j,k)$ be the cent-based log frequency of the kth pitch candidate of the jth note in the humming. The deltaPitches are

$$\Delta_1 h_p(j,k,l) = h_p(j+1,k) - h_p(j,l) \tag{36}$$
$$\Delta_2 h_p(j,k,l) = h_p(j+1,k) - h_p(j-1,l). \tag{37}$$

Then, the distance between deltaPitches are defined as

$$p_1(i,j,k,l) = |\Delta_2 m_p(i) - \Delta_1 h_p(j,k,l)| \tag{38}$$
$$p_2(i,j,k,l) = |\Delta_1 m_p(i) - \Delta_1 h_p(j,k,l)| \tag{39}$$
$$p_3(i,j,k,l) = |\Delta_1 m_p(i) - \Delta_2 h_p(j,k,l)|. \tag{40}$$

Next, let $h_c(j,k)$ be the confidence measure of the kth pitch candidate of the jth note in the humming. Now, the sums of confidence measures c_1, c_2 and c_3 are calculated as

$$c_1(j,k,l) = c_2(j,k,l) = h_c(j+1,k) + h_c(j,l) \tag{41}$$
$$c_3(j,k,l) = h_c(j+1,k) + h_c(j-1,l). \tag{42}$$

The distances d_e (e = 1,2,3) are then calculated as

$$d_e(i,j,k,l) = \beta\left\{\alpha p_e(i,j,k,l) + (1-\alpha)c_e^{-1}(j,k,l)\right\} + (1-\beta)t_e(i,j). \tag{43}$$

The factors α and β can be varied to reflect the relative contribution of pitch, confidence measure and IOI. When $\alpha = 1$ and $\beta = 1$, the weight contribution is only based on pitch. On the other hand, if the β value is zero, the weight contribution is only based on IOI.

Finally, the DP distance between the input and reference sequences is calculated as follows:

$$g(i,-1,k) = \infty \quad (0 \le i \le I-1) \tag{44}$$
$$g(-1,j,k) = \infty \quad (0 \le j \le J-1) \tag{45}$$
$$g(i,0,k) = \min_l d_2(i,0,k,l) \quad (0 \le i \le I-1) \tag{46}$$
$$g(0,j,k) = \infty \quad (1 \le j \le J-1) \tag{47}$$
$$g(i,j,k) = \min \begin{cases} \min_l \{g(i-2,j-1,l) + d_1(i,j,k,l)\} \\ \min_l \{g(i-1,j-1,l) + d_2(i,j,k,l)\} \\ \min_l \{g(i-1,j-2,l) + 2d_3(i,j,k,l)\} \end{cases}. \quad (0 \le k \le K-1, 0 \le l \le K-1) \tag{48}$$

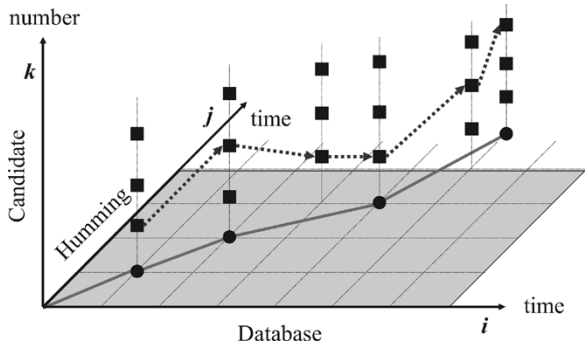

Fig. 7. Example of the matching flow using the 3-D continuous DP algorithm.

TABLE II
Experimental Conditions

Music Database	155 pieces of Children's songs from Japan and other countries
Test Data	Humming by 5 subjects
Sampling Frequency	16 kHz
Window size	64 ms (Hamming Windows)
frame shift	8 ms
Event Detection	BPF: 600-1,500 Hz DF: Primary differential
Features	Multiple pitch candidates Confidence measure IOI

(BPF: Band Pass Filter, DF: Differential Filter)

where K stands for the maximum number of the pitch candidate of one note. Fig. 7 shows the mechanism of the matching algorithm extended to three dimensions. The humming is matched with the database at the DP plane extended to three dimensions. When considering the matching of the humming to the database, the proposed algorithm calculates the combination of all candidate points [20]. Finally, the algorithm determines the optimal candidate points and paths, as shown in Fig. 7.

V. Experiments and Results

A. Experimental Conditions

The music database contained 155 children's songs from Japan and other countries. Each song in the database is represented as monophonic MIDI data that contains the melody line only. The pitches and the IOIs were extracted from the monophonic MIDI. We used 320 pieces of hummed query data from five subjects for the experiments. All subjects were inexperienced singers. They were instructed that they could start the songs from an arbitrary position. There were no restrictions about the tonality and tempo of the humming.

Table II shows the experimental conditions. In the experiment, the last note was neglected because it has no deltaPitch and IOI. The average number of notes in one humming data was 10.7 and the average humming time was 5.3 s. In the experiment, weighted values (α, β) were changed from 0 to 1 by 0.1.

B. Evaluation Measurement

The result of the retrieval is evaluated by the *retrieval accuracy*. The retrieval accuracy $A(R)$ is the probability that the target song is included in the top R outputs. Many retrieval systems display the top R outputs for a query. Those systems often select exactly R outputs from the list even if the $(R+1)$th output has the same score as the Rth output.

TABLE III
EXAMPLE OF THE LIST OF RETRIEVAL RESULTS

Ranking	Title of song	Scores
1	ookinahurudokei	98.1
2	chiisaiakimituketa	95.3
2	hosinosekai	95.3
2	harunokaze	95.3
5	makkanaaki	87.4
6	hureru	82.6

TABLE IV
DYNAMIC THRESHOLD VALUES FOR DELTAPITCH/IOIRATIO IN CATEGORY 3

Dynamic threshold in IOI		Dynamic threshold in deltaPitch	
$S < 93.7$	Shorter	$P < -147$	Down
$93.7 \leq S < 109.8$	Equal	$-147 \leq P < 148$	Same
$109.8 \leq S$	Longer	$148 \leq P$	Up

Therefore, the retrieval accuracy takes that behavior into account. The retrieval accuracy $A(R)$ was calculated as follows:

$$A(R) = \frac{1}{Q}\sum_{i=1}^{Q} T_i(R). \tag{49}$$

$$T_i(R) = \begin{cases} 1, & \text{if } r(i) + n_i\left(r(i)\right) - 1 \leq R, \\ 0, & \text{else if } r(i) > R, \\ \frac{R - r(i) + 1}{n_i(r(i))}, & \text{otherwise} \end{cases} \tag{50}$$

where Q denotes the number of queries and $n_i(R)$ is the number of candidates that have the same score as the Rth candidate in the ith query outputs. $r(i)$ is the rank of the correct candidate among the retrieval outputs of the ith query. $T_i(R)$ means the probability that the target song is included in the top R outputs of the ith query exactly, assuming that the order of the candidates with the same score in the output list is determined randomly.

Table III shows an example of the list of retrieval results. When the target song is *hosinosekai*, $r(i) = 2$, $n_i(2) = 3$ and $A(3) = 2/3$.

C. Results

In order to investigate the performance of the proposed algorithm, we carried out music retrieval experiments. In the evaluation, a conventional MIR system was implemented and the retrieval results were compared with those of the proposed system. The pitch obtained by YIN was also compared.

The Coarse-to-Fine Matching method [6] was used to effectively reduce the number of answering candidates by gradually increasing the number of categories of relative-values used in the DP matching. For example, three categories of pitch represent the situation where a note is above (up), below (down) or equal (same) to the previous note. In the experiment, the Coarse-to-Fine matching utilized three steps to increase the categories $(3 \rightarrow 9 \rightarrow 27)$.

Boundaries of the categories in this system were tuned using all of the humming data. Table IV shows the boundary for pitch and IOI obtained from the data. The deltaPitch values are represented in cents, and the IOIratio values are represented as a percentage.

The experimental results for music retrieval are shown in Table V. Here, "Coarse-to-Fine" and "Category 27" denote the methods in reference [6], and "YIN Pitch" denotes the retrieval results when the pitch value was obtained using [24]. "Conventional" denotes the static melody representation described in Section II-A. The γ in Table V is the weighting factor in (7).

From these results, the proposed melody representation gave about three to seven points higher retrieval accuracies compared to the conventional method. By using multiple pitch candidates, the retrieval accuracy improved from 73.9% to 86.5%. This result outperformed the result using YIN-based pitch extraction. These results are consistent with the results of the pitch extraction accuracy shown in Table I. Compared to "Category 27", the proposed method showed a higher accuracy of about a five points.

TABLE V
COMPARISONS OF THE ACCURACY BY VARIOUS FEATURES (%)

Features			Melody representation				Weight values	
			Conventional		Proposed			
IOI	Pitch	CM	$A(1)$	$A(10)$	$A(1)$	$A(10)$	α	β
√			40.9	67.3	42.6	71.2	—	0.0
	1		61.4	83.9	66.1	90.4	1.0	1.0
√	1		70.8	86.7	73.9	91.1	1.0	0.6
√	3	√	79.6	90.2	86.5	94.1	0.5	0.7
Coase-to-Fine			78.4	89.6			$\gamma = 0.5$	
Category 27			81.6	91.5			$\gamma = 0.5$	
YIN Pitch					83.7	89.7	$\gamma = 0.5$	

(CM:Confidence Measure)

VI. CONCLUSIONS

In this paper, we have proposed a novel error-tolerant melody-matching method for retrieving music information in response to hummed queries. The user's hummed input differs from the ideal input for several reasons, including individual differences in tonality and tempo, and singing errors based on ambiguous memory. Considering these problems, the optimum neighboring note is dynamically determined to take the relative-value according to the DP paths.

Furthermore, even if the hummed queries are perfect, it is still difficult to retrieve the pitch perfectly from the hummed queries. To consider the pitch extraction errors, we proposed the use of multiple pitch candidates. Using the proposed method, a pitch extraction accuracy of 99.7% was obtained within the third rank. Moreover, we proposed the use of a similarity measurement algorithm that extends the search space of the DP plane into three dimensions for robust matching to the pitch extraction errors in the query processing. This is based on a continuous dynamic programming algorithm with features that include IOIs and multiple pitch candidates along with their confidence measures.

We evaluated this system by measuring retrieval accuracy on a database of 155 songs with a total of 320 queries. When using three-pitch candidates with confidence measures and IOI features, the top-ten retrieval accuracy was 94.1% and the top-one retrieval accuracy was 86.5%. These results showed better retrieval performance than the conventional system. Thus, the advantage of the proposed method is apparent.

REFERENCES

[1] D. Feng, W. C. Siu, and H. Zhang, *Multimedia Information Retrieval and Management: Technological Fundamentals and Applications*. Berlin, Germany: Springer-Verlag, 2003.
[2] J. T. Foote, "A similarity measure for automatic audio classification," in *Proc. AAAI 1997 Spring Symp. Intelligent Integration and Use of Text, Image, Video, and Audio Corpora*, Stanford/Palo Alto, CA, Mar. 1997.
[3] A. R. Coden, E. W. Brown, and S. Srinivasan, *Information Retrieval Techniques for Speech Applications (LNCS 2273)*. Berlin, Germany: Springer-Verlag, 2002.
[4] T. Kageyama, K. Mochizuki, and Y. Takashima, "Melody retrieval with humming," in *Proc. Int. Computer Music Conference*, 1993.
[5] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith, "Query by humming: musical information retrieval in an audio database," in *Proc. ACM Multimedia*, 1995.
[6] T. Sonoda, M. Goto, and Y. Muraoka, "WWW-based music retrieval system," in *Proc. ICMC'98*, 1998, pp. 343–352.

[7] T. Shirokaze, S. Makino, and K. Kido, "Extraction of fundamental frequency using temporal continuity over an input speech," *Trans. IEICE*, vol. 73-A, no. 9, pp. 1537–1539, 1990.
[8] A. Klapuri, "Pitch estimation using multiple independent time-frequency windows," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 1999, pp. 115–118.
[9] C. Wei and V. Barry, "Folk music classification using hidden Markov models," in *Proc. Int. Symp. Music Information Retrieval*, Oct. 2000.
[10] S. Pauws, "CubyHum: a fully operational query by humming system," in *Proc. ISMIR 2002*, 2002.
[11] L. R. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[12] S.-P. Heo, M. Suzuki, A. Ito, S. Makino, and H. Chung, "Multiple pitch candidates based music information retrieval method for query-by-humming," in *Int. Workshop AMR2003*, 2003, pp. 189–200.
[13] J. Paulus and A. Klapuri, "Measuring the similarity of rhythmic patterns," in *Proc. ISMIR 2002*, 2002.
[14] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Sunningham, "Toward the digital music library: tune retrieval from acoustic input," in *Proc. ACM Digital Libraries*, Bethesda, MD, 1996.
[15] Themefinder, Stanford Univ.. Stanford, CA [Online]. Available: http://www.themefinder.org/
[16] TuneServer, Univ. Karlsruhe, Germany [Online]. Available: http://name-this-tune.com/
[17] MiDiLiB, Univ. Bonn, Germany [Online]. Available: http://www-mmdb.iai.uni-bonn.de/forschungprojekte/midilib/english/
[18] J. S. R. Jang, H. Lee, and J. Chen, "Super MBox: an efficient/effective content-based music retrieval system," in *Ninth ACM Multimedia Conf. (Demo Paper)*, 2001, pp. 636–637.
[19] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamuro, and K. Kushima, "A practical query-by-humming system for a large music database," in *ACM Multimedia 2000*, 2000, pp. 333–342.
[20] S.-P. Heo, M. Suzuki, A. Ito, and S. Makino, "Three dimensional continuous DP algorithm for multiple pitch candidates in music information retrieval system," in *Proc. ISMIR 2003*, 2003.
[21] R. J. McNab, L. A. Smith, and I. H. Witten, "Signal processing for melody transcription," in *Proc. 19th Australasian Computer Science Conf.*, 1996.
[22] T. Nishimura, H. Hashiguchi, J. Takita, J. X. Zhang, M. Goto, and R. Oka, "Music signal spotting retrieval by a humming query using start frame feature dependent continuous dynamic programming," in *Proc. ISMIR 2001*, 2001, pp. 211–218.
[23] H. Hashiguchi, T. Nishimura, J. Takita, J. X. Zhang, and R. Oka, "Music signal spotting retrieval by a humming query," in *SCI 2001*, 2001, vol. VII, pp. 280–284.
[24] A. de Cheveigne and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *J. Acoust. Soc. Am.*, vol. 111, pp. 1917–1930, 2002.
[25] P. Boersma and D. Weenink, *Praat*. Amsterdam, The Netherlands: Univ. Amsterdam Press [Online]. Available: http://www.fon.hum.uva.nl/praat/