# Music Signal Spotting Retrieval by a Humming Query Using Start Frame Feature Dependent Continuous Dynamic Programming

Takuichi Nishimura
Real World Computing Partnership
/ National Institute of Advanced
Industrial Science and Technology
Tsukuba Mitsui Building 13F, 1-6-1
Takezono Tsukuba-shi, Ibaraki
305-0032, Japan
+81-298-53-1686
nishi@rwcp.or.jp

Hiroki Hashiguchi
Real World Computing Partnership
Tsukuba Mitsui Building 13F, 1-6-1
Takezono Tsukuba-shi, Ibaraki
305-0032, Japan
+81-298-53-1668
hiro@rwcp.or.jp

Junko Takita
Mathematical Systems Inc.
2-4-3 Shinjuku, Shinjuku-ku, Tokyo
160-0022 Japan
+81-3-3358-1701
takita@msi.co.jp

J. Xin Zhang
Media Drive Co.
3-195 Tsukuba, Kumagaya-shi,
Saitama, 360-0037, Japan
+81-48-524-0501
chou@mediadrive.co.jp

Masataka Goto
National Institute of Advanced
Industrial Science and Technology
/"Information and Human Activity"
PRESTO, JST
1-1-1 Umezono, Tsukuba-shi, Ibaraki
305-8568, Japan
+81-298-61-5898
m.goto@aist.go.jp

Ryuichi Oka
Real World Computing Partnership
Tsukuba Mitsui Building 13F, 1-6-1
Takezono Tsukuba-shi, Ibaraki
305-0032, Japan
+81-298-53-1686
oka@rwcp.or.jp

## ABSTRACT

We have developed a music retrieval method that takes a humming query and finds similar audio intervals (segments) in a music audio database. This method can also address a personally recorded video database containing melodies in its audio track. Our previous retrieving method took too much time to retrieve a segment: for example, a 60-minute database required about 10-minute computation on a personal computer. In this paper, we propose a new high-speed retrieving method, called *start frame feature dependent continuous Dynamic Programming*, which assumes that the pitch of the interval start point is accurate. Test results show that the proposed method reduces retrieval time to about 1/40 of present methods.

## 1. INTRODUCTION

Although a large amount and variety of musical audio signals have been available on the internet and stored in personal hard disks at home, information retrieval methods for those signals are still in infancy. A typical method is a simple text-based search which seeks property tags attached to those signals, such as the song title, artist name, and genre. The purpose of this research is to enable a user to retrieve a segment of a musical audio signal desired just by singing its melody. Such a music retrieval method is also useful for retrieving video clips if those clips contain music in audio tracks. For practical application, we think it important that the method can be applied to an audio database that is not segmented into musical pieces (as is often the case with broadcast recordings).

In this paper, we focus on a music retrieval system that takes a humming query and finds similar audio intervals (segments) in a music audio database. If the database is composed of melody scores such as MIDI, symbols (e.g., relative pitch change or span change) can be extracted robustly. In this case, symbol-based retrieval [1-4] is very efficient. On the other hand, extracted melody from an audio signal usually suffers a lot of error and such symbol-based methods are not applicable. Therefore, we developed the pattern-based retrieval method [5]. Fundamentally, we find a similar pitch sequence of a query (query pattern) in the melody-likeness pattern on the pitch-temporal plane obtained from the database shifting the query pattern along pitch axis and warping temporally as shown in Figure 1. The previous matching method is called *model-driven path Continuous Dynamic Programming (mpCDP)* which compares the reference and all the partial intervals in the database and outputs similar intervals by considering multiple possibilities of transpositions. Because the mpCDP achieves pattern-based matching, the method can also take whistle sound or tempo-varying humming. (In the following, we use two terms, "reference pattern" and "input pattern": we extract a "reference pattern" from a query and an "input pattern" from a database in order to feed those patterns into matching methods.) This previous mpCDP, however, is too slow to be used in practical applications: it needs much computational cost because it accumulates local similarities in 3-D space, which is composed from model-axis, input-axis, and pitch-axis.

In this paper, we propose a new quick retrieval method, called *start frame feature dependent continuous DP (s-CDP),* which searches only in 2-D space (reference-axis and input-axis) assuming that the feature (pitch in this paper) of the start point of the extracted optimal interval is accurate. Our new s-CDP is different from conventional continuous DP in the respect that we do not calculate local similarities beforehand because the start point of the optimal interval is obtained from bottom left to top right successively in the 2-D space.

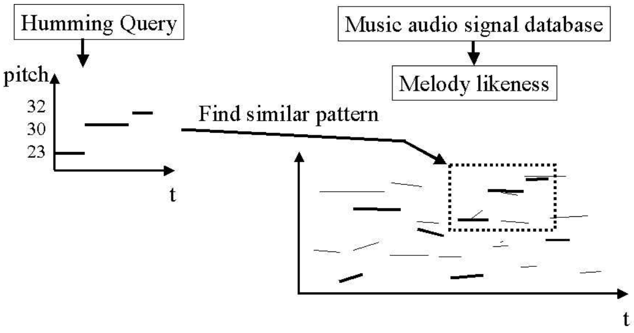


**Figure 1. Pattern matching for humming retrieval. Pitch shift and temporal-warp are considered.**

We evaluate our new method using 20 popular music selections comparing the conventional method to show that the proposed method can reduce search time to about 1/40 with retrieval rates in excess of 70%.

The following section describes the conventional retrieval method. The s-CDP and the retrieval method using s-CDP is proposed in Section 3. The method is evaluated in Section 4 and newly developed retrieval system is introduced in Section 5.

## 2. OUR CONVENTIONAL RETRIEVAL METHOD

Continuous DP [7] is improved from temporally monotonous Dynamic Programming (DP) for speech or gesture recognition. Continuous DP achieves inconsistent recognition because it can segment the input pattern automatically but it cannot consider multiple possibilities of transpositions. Therefore, we proposed mpCDP [5] for music retrieval method adding one dimension (pitch) for input pattern and finding similar pitch sequence to the query.

### 2.1 Continuous DP

As shown in Figure 2 (a), continuous DP calculates accumulated similarities $S(t,\tau)$ between reference $R_\tau$ $(1 \le \tau \le T)$ and input $I_t$ $(0 \le t \le \infty)$ by adding local similarities $s(t,\tau)$. Denoting the temporal axis of reference and input as $t$ and $\tau$ , respectively, continuous DP calculates $S(t,\tau)$ using the following recursive equations.

*Boundary conditions* $(1 \leq \tau \leq T, 0 \leq t)$:

$$S(t,0) = S(t,-1) = 0$$

$$S(-1,\tau) = S(0,\tau) = 0$$

*Recursive equations* $(1 \leq t)$:

$$S(t,1) = 3 \cdot s(t,1)$$

$$S(t,\tau) =$$

$$\max \begin{cases} S(t-2,\tau-1) + 2s(t-1,\tau) + s(t,\tau) \\ S(t-1,\tau-1) + 3s(t,\tau) \\ S(t-1,\tau-2) + 3s(t,\tau-1) + 3s(t,\tau) \end{cases} \quad (1)$$

$$(2 \leq \tau \leq T)$$

In this equation, local path with maximum similarity is chosen among the three paths as shown in Figure 2(c). Numbers besides each point are weights for local similarities. Notice that the summation of weights is always three for one frame up along the reference axis. Therefore, dividing accumulated similarities by three can normalize them.

Next, we find a similar interval. Figure 2(b) shows accumulated similarities $S(t,T)$; and the continuous DP decide the maximum point with higher similarity than a certain threshold $\alpha$ as the end point of the similar interval.

In brief, continuous DP finds the optimal path and maximum accumulated similarities by choosing the maximum local path successively from bottom left to top right in the reference-input plane. The $S(t,T)$ is the similarity between the whole reference and the input considering temporal warps from 1/2 to 2 times.

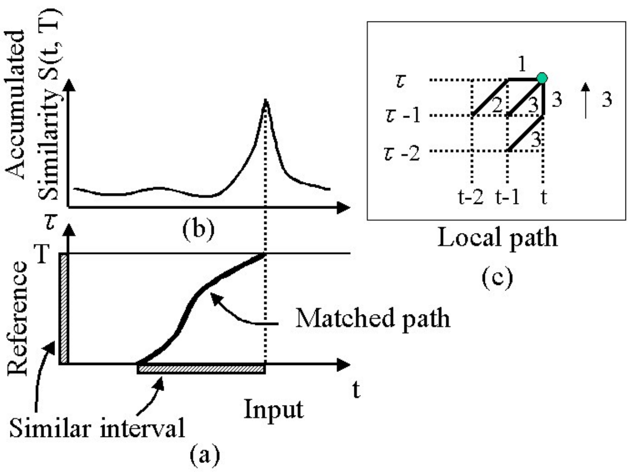


**Figure 2. Continuous Dynamic Programming.**

## 2.2 Humming Retrieval by mpCDP

This section explains the conventional humming retrieval method using Figure 3. The method has three steps. First, database audio signals are analyzed every frame, 64ms in this paper, and sequence of melody-likeness of each pitch (SMLP) is obtained. When a query is input to the method, highest melody-likeness pitch is chosen from SMLP every frame and query model is created by the relative pitch change. Third, model-driven path Continuous Dynamic Programming (mpCDP) compares the model and all partial intervals in the database and output similar intervals by considering multiple possibilities of transpositions.

The local path of mpCDP is different from that of continuous DP in that they are shifted along the pitch axis according to the model, which is the relative pitch change of the query. By executing such processes successively, accumulated similarity $S(t,T,x)$ (Here $x$ denotes pitch axis) takes a maximum similarity along the pitch axis at the end of the model as shown in top-right figure of Figure 3. Then, $\max_x S(t,T,x)$ changes similar to Figure 2(b). Finally the mpCDP outputs the maximum point with higher similarity than a certain threshold $\alpha$ as the end point of the similar interval.

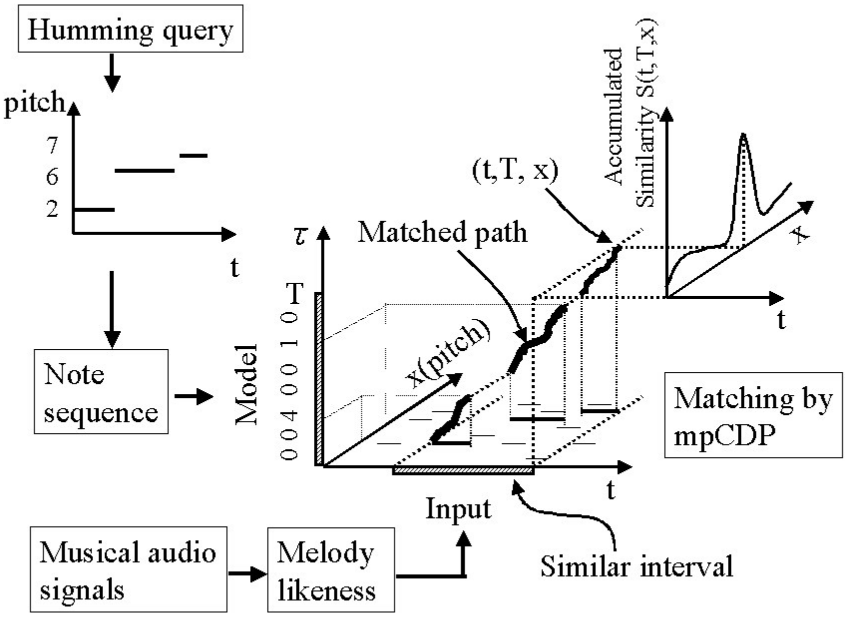
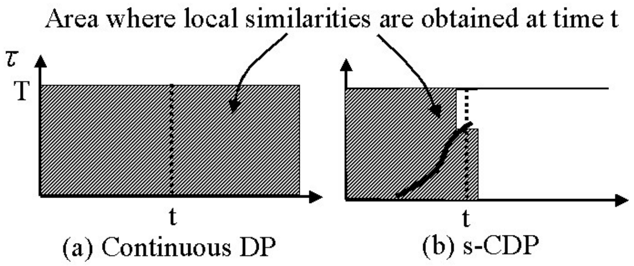


**Figure 3. Conventional retrieval method.**

## 3. PROPOSAL OF s-CDP

### 3.1 s-CDP

The definition of s-CDP is that local similarities $s(t,\tau)$ are dependent on the feature of the start point $(p(t,\tau),1)$ of optimal paths. Therefore local similarities of s-CDP are described as $s(t,\tau)=f(R_\tau,I_t,R_1,I_{p(t,\tau)})$ using a certain similarity function $f()$ . (For continuous DP: $s(t,\tau)=f(R_\tau,I_t)$).



(a) Continuous DP

(b) s-CDP

**Figure 4. Comparison of matching methods.**

The difference between conventional continuous DP and s-CDP is shown in Figure 4. All local similarities can be calculated beforehand for continuous DP (Figure 4(a)), whereas local similarities are obtained incrementally as the optimal paths are fixed for s-CDP as shown in Figure 4(b). There are three local similarities for each point $(t,\tau)$ because the start points are different among three local paths as shown in Figure 2 (c). Equation (1) is rewritten for s-CDP as:

*Recursive equations* ($1 \le t$):

$$S(t,1) = 3 \cdot s_2(t,1)$$

$$S(t,\tau) =$$

$$\max\begin{cases} S(t-2,\tau-1) + 2s_2(t-1,\tau) + s_1(t,\tau) \\ S(t-1,\tau-1) + 3s_2(t,\tau) \\ S(t-1,\tau-2) + 3s_2(t,\tau-1) + 3s_3(t,\tau) \end{cases} \quad (2)$$

$$(2 \le \tau \le T)$$

Here we call the local paths in Figure 2 (c) path1, path2, path3 from top-left and define $s_1(t,\tau)$, $s_2(t,\tau)$, $s_3(t,\tau)$ as local similarities for path1, path2, and path3, respectively. Those are defined as:

$$s_2(t,\tau) = f(R_\tau, I_t, R_1, I_{p(t,\tau)}) \ (\tau = 1)$$

$$s_1(t,\tau) = f(R_\tau, I_t, R_1, I_{p(t-2,\tau-1)}) \ (2 \le \tau)$$

$$s_2(t,\tau) = f(R_\tau, I_t, R_1, I_{p(t-1,\tau-1)}) \ (2 \le \tau)$$

$$s_3(t,\tau) = f(R_\tau, I_t, R_1, I_{p(t-1,\tau-2)}) \ (2 \le \tau)$$

The $s_2(t,\tau)$ alone takes $\tau = 1$ and requires exception because the start point is the same as the point $(t,1)$. Second terms of path1 and path3 in equation (2) are $s_2(\cdot,\cdot)$ because both points come from points (–1,–1) relatively in the plane.

Outputs of s-CDP are obtained as points with maximum accumulated similarities in the same way as continuous DP, but they have positions of start points and features of start points.

To calculate the start position $p(t,\tau)$ on the input axis, first initialize with time $t$ when $\tau = 0,1$.

$$p(t,0) = p(t,1) = t$$

Next, copy the start point memorized at the local start point to $p(t,\tau)$ as follows:

$$p(t,\tau) = \begin{cases} p(t-2,\tau-1) \ (\mathit{if}\ \ path1) \\ p(t-1,\tau-1) \ (\mathit{if}\ \ path2) \\ p(t-1,\tau-2) \ (\mathit{if}\ \ path3) \end{cases}$$

Start positions $(p(t,\tau),1)$ are obtained incrementally by the recursive equations above.

In case of error in start points, features of start points $R_1, I_{p(t,\tau)}$ are incorrect - leading to miscalculation of accumulated similarities. Therefore, one must choose a high melody-likeness point as the start point in order to segment the reference. As for input, segmenting a similar interval is the method's purpose, so such a measure is impossible. Still, those errors are recovered if one feature near the start point is correct because s-CDP matches, warping the reference temporarily.

### 3.2 Apply for Humming Retrieval

In this section, s-CDP is modified for the humming retrieval method. Because melody is expressed as the pitch sequence, the feature is pitch and each pitch is subtracted from the start pitch to cope with transpositions, assuming that start pitch is correct. Query is transposed to make the start pitch 0 as shown in Figure 5. On the other hand, database transposition is possible just after the start pitch is fixed by tracing back the optimal path derived from s-CDP. Then the segmented database is similarly transposed to make the start pitch 0 as shown in Figure 5. Finally, s-CDP compares the transposed query and database without any influence of transpositions.

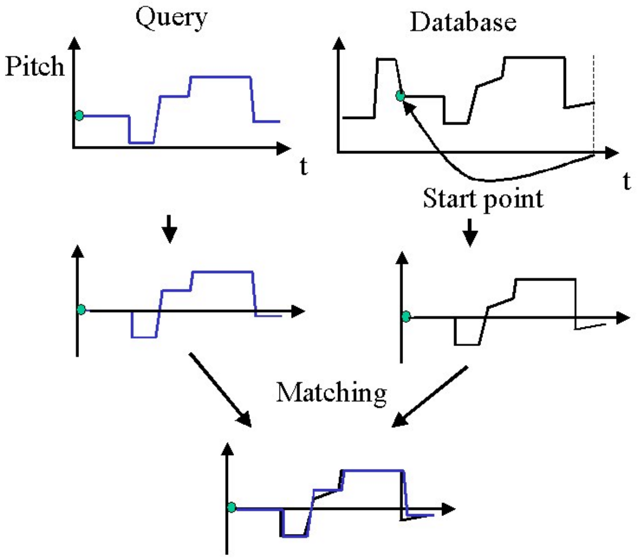


**Figure 5. Overview of melody matching considering transpositions.**

Figure 6 helps to explain the detail method. First, make relative pitch sequence of query as the reference $R_\tau^{'} = R_\tau - R_1$. Second, obtain N high melody-likeness candidates from database musical signal as input $I_t(k)(k = 1,\cdots,N)$. In this study, local similarities $s(t,\tau)$ are defined:

$$s(t,\tau) = \begin{cases} 1(D_{t,\tau} = 0) \\ 0(D_{t,\tau} \neq 0) \end{cases}$$

Here $D_{t,\tau} = \min_k abs[R_\tau^{'} - \{I_t(k) - I_{p(t,\tau)}(1)\}]$ and local similarities are 1 if one of the relative pitches of N candidates is equal to relative pitch of the query. We set $N = 5$ in experiments in this paper.

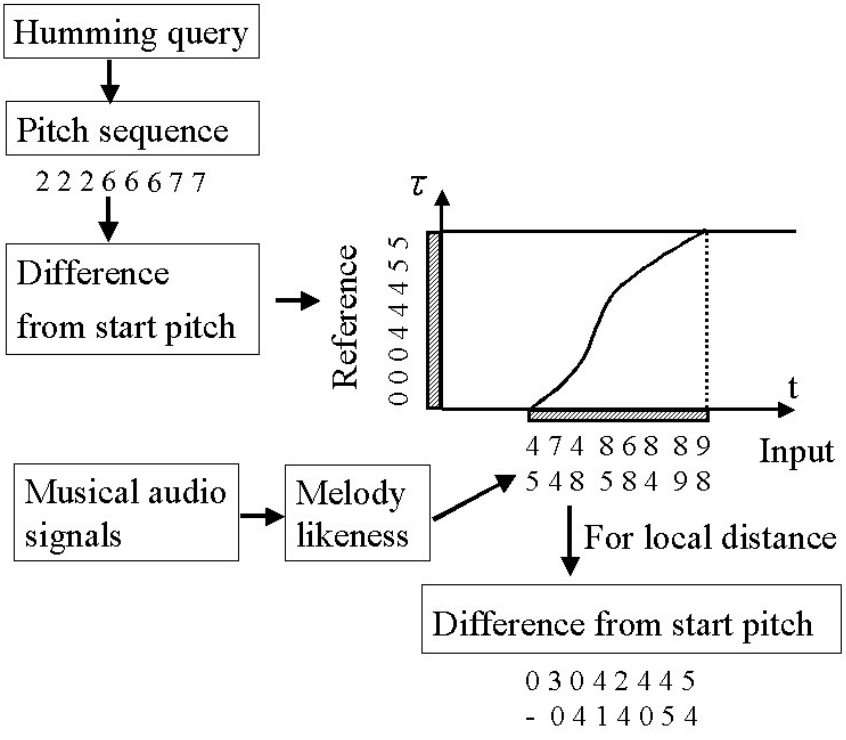


**Figure 6. Proposed retrieval method using s-CDP.**

# 4. EXPERIMENTS

## 4.1 Experimental Method

We prepared 20 WAV files (about 80 minutes in total) in 16 kHz sampling and monaural recording format as a music database to compare s-CDP with mpCDP. This database includes 10 Japanese pop songs, 8 children's songs, an animation song, and a Japanese *enka*. There are 4 male and 16 female vocal artists in the database. Also, 3 males and 2 females sang a portion of each song in the database for about 20 seconds. Hence, the total number of queries was 100.

Let $f_b$[Hz] (in this experiment: 55[Hz]) denote the lowest frequency of melody. We compute the melody likeness for the frequency $2^{x/12} f_b (x=1,\cdots,X)$ [Hz] by FFT analysis in consideration of the harmonic structure of acoustic signals. This method is a simple version of [5]. The step of the pitch axis of mpCDP is set at 60 (5 octaves: $X=60$) considering the male and female voice pitch range.

Two thresholds segmented a reference from a query. To decide the start point, the threshold is set at $0.5\overline{P}$, where $\overline{P}$ is average power (summation of square of signal). For the end point, the threshold is set at $0.1\overline{P}$. The start point threshold is set larger because the start point pitch should be correct.

The search rate, which depends on a threshold $\alpha(0 \le \alpha \le 1)$, is defined as the average of precision rate *NC/ND* and recall rate *NC/NT*, where *NC*, *ND,* and *NT* represent the number of similar terms to the humming query in detected terms, the number of detected terms by mp-CDP, and the number of similar terms to the humming, respectively. The detected term by mp-CDP is correct if the following overlapping rate

$$\text{overlapping rate} = \frac{\text{similar term} \cap \text{detected term}}{\text{similar term} \cup \text{detected term}}$$

is greater than 0.5. This means the overlapping terms between similar and detected ones has 70% intersection when lengths of both terms are mutually identical. Since the search rate depends on threshold $\alpha$, "the search rate for a humming query" is defined as the maximum value running over all $\alpha$. The computer for this experiment is OS: Windows2000, CPU: Pentium IV 1.5GHz.

## 4.2 Results

Table 1 shows average search rates and search times for 5 persons × 20 songs = 100 queries for query duration of 20 seconds. Those results show that s-CDP reduced search time to about 1/40. The mpCDP has 60 times the local path calculation because it has 60 steps in pitch axis. The reduction effect is only about 1/40 because s-CDP calculates the start point and has three times the number of local similarities.

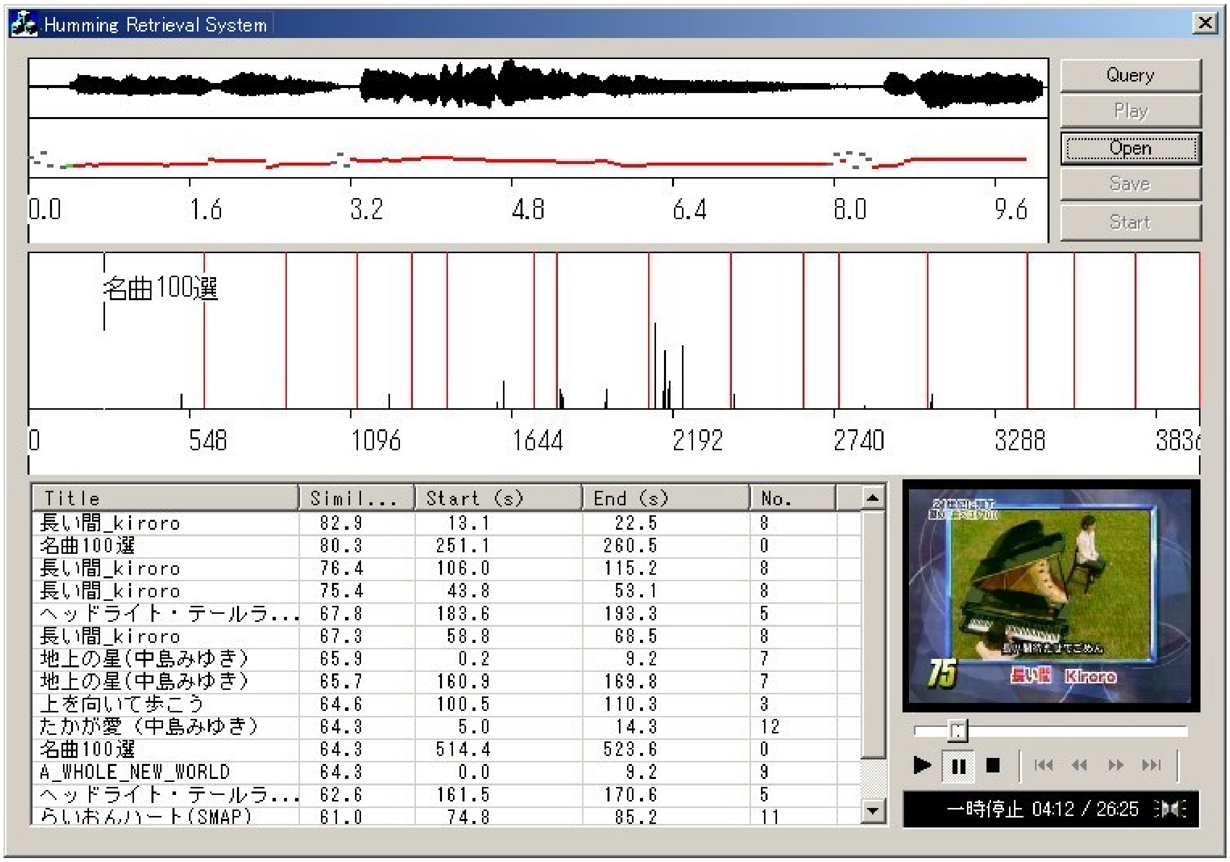
The s-CDP search rate was lower by 16%, though it is more than 70%, showing practical usefulness.

Table 1: Comparison of the two search methods.

| Search method | mpCDP | s-CDP |
|---|---|---|
| Average search rate | 89.0% | 73.3% |
| Search time | 532(s) | 14.2(s) |

## 5. RETRIEVAL SYSTEM

We made a humming retrieval system as shown in Figure 7. From the top window, the query wave, the query pitch sequence, similarities in the database, and hit results are shown. On clicking the peak on the similarities or the hit results, similar intervals of video or music are played in the right-bottom window. Using a notebook PC with a Pentium III 750MHz CPU, retrieving a 10[s] query from 60[min] database took about 10 seconds. There are several similarity peaks in Figure 7 because the same song was recorded from a TV program and a radio program and also the song had several repeated melodies. Retrieving from an audio video database that captured a scene from karaoke has been successful with this system, which will be demonstrated in presentation.



| Title | Simil... | Start (s) | End (s) | No. |
|---|---|---|---|---|
| 長い間_kiroro | 82.9 | 13.1 | 22.5 | 8 |
| 名曲100選 | 80.3 | 251.1 | 260.5 | 0 |
| 長い間_kiroro | 76.4 | 106.0 | 115.2 | 8 |
| 長い間_kiroro | 75.4 | 43.8 | 53.1 | 8 |
| ヘッドライト・テールラ... | 67.8 | 183.6 | 193.3 | 5 |
| 長い間_kiroro | 67.3 | 58.8 | 68.5 | 8 |
| 地上の星(中島みゆき) | 65.9 | 0.2 | 9.2 | 7 |
| 地上の星(中島みゆき) | 65.7 | 160.9 | 169.8 | 7 |
| 上を向いて歩こう | 64.6 | 100.5 | 110.3 | 3 |
| たかが愛（中島みゆき） | 64.3 | 5.0 | 14.3 | 12 |
| 名曲100選 | 64.3 | 514.4 | 523.6 | 0 |
| A_WHOLE_NEW_WORLD | 64.3 | 0.0 | 9.2 | 9 |
| ヘッドライト・テールラ... | 62.6 | 161.5 | 170.6 | 5 |
| らいおんハート(SMAP) | 61.0 | 74.8 | 85.2 | 11 |

**Figure 7. Monitor of the developed retrieval system.**

## 6. SUMMARY

We proposed a start frame feature dependent continuous DP assuming that the start point pitch of the extracted optimal interval is correct. Test results showed that the proposed method reduces computational costs to about 1/40.

One method for further reducing retrieval time is to compress similar intervals in the database because a song usually has a repeated melody.

## 7. REFERENCES

[1] Kageyama T., Mochizuki K., and Takashima Y.: Melody Retrieval with Humming, ICMC Proc., 1993, 349-351.

[2] Asif Ghias and Logan J.: Query By Humming – Musical Information Retrieval in an Audio Database, ACM Multimedia '95, Electronic Proc., 1995.

[3] Sonoda T., Goto M., and Muraoka Y.: A WWW-based Melody Retrieval System, ICMC'98 Proc., 1998, 349-352.

[4] Kosugi N., Nishihara Y., Sakata T., Yamamuro M., and Kushima K.: A practical Query-by-Humming system for a large music database, ACM Multimedia 2000, 333--342.

[5] Hashiguchi H., Nishimura T., Takita J., Zhang J. X., and Oka R.: Music Signal Spotting Retrieval by Humming Query Using Model Driven Path Continuous Dynamic Programming, SCI2001,

[6] Goto M.: A Predominat-F0 Estimation Method for CD Recordings: MAP Estimation using EM Algorithm for Adaptive Tone Models, Proc. of ICASSP 2001.

[7] Oka R.: Continuous word recognition with Continuous DP (in Japanese), Report of the Acoustic Society of Japan, S78-20, 1978, 145-152.