

Using the Similarity of Main Melodies to Identify Cover Versions of Popular Songs for Music Document Retrieval*

WEI-HO TSAI, HUNG-MING YU⁺ AND HSIN-MIN WANG⁺

*Department of Electronic Engineering
National Taipei University of Technology
Taipei, 106 Taiwan*

*⁺Institute of Information Science
Academia Sinica
Taipei, 115 Taiwan*

Automatic extraction of information from music data is an important and challenging issue in the field of content-based music retrieval. As part of the research effort, this study presents a technique that automatically identifies cover versions of songs specified by users. The technique enables users to search for songs with an identical tune, but performed by different singers, in different languages, genres, and so on. The proposed system takes an excerpt of the song specified by the user as input, and returns a ranked list of songs similar to the input excerpt in terms of the main melody. To handle likely discrepancies, *e.g.*, in tempo, transposition, and accompaniment, between cover versions and the original song, methods are presented to remove the non-vocal portions of the song, extract the sung notes from the accompanied vocals, and compare the similarities between the sung note sequences. Our experiments on a database of 594 cross-lingual popular songs show the feasibility of identifying cover versions of songs for music retrieval.

Keywords: content-based music retrieval, cover version, main melody, polyphonic, accompaniments

1. INTRODUCTION

Recent advances in digital signal processing technologies, coupled with what are essentially unlimited data storage and transmission capabilities, have created an unprecedented growth in the amount of music material being produced, distributed, and made available universally. At the same time, our ever-increasing appetite for music has provided a major impetus to the development of various new technologies. However, as the amount of music-related data and information continues to grow, finding a desired item among the innumerable options can, ironically, be more difficult. This problem has motivated research into developing techniques for automatically extracting information from music. Specific topics, such as music structural analysis [1-3], melody spotting [4, 5], beat tracking [6, 7], music transcription [8, 9], genre classification [10, 11], singer identification [12, 13], instrument recognition [14, 15], and audio fingerprinting [16, 17],

Received February 27, 2007; revised August 17, 2007; accepted October 18, 2007.

Communicated by Jorng-Tzong Horng.

* This paper was partially supported by the National Science Council of Taiwan, R.O.C. under grants No. NSC 93-2422-H-001-0004, NSC 94-2422-H-001-007, NSC 95-2422-H-001-008, and NSC 95-2218-E-027-020. Part of this paper has been presented in the International Conference on Music Information Retrieval, Sept. 11-15, 2005, London, UK, Queen Mary, University of London.

are being extensively studied within the overall context of content-based music retrieval [18-23]. In tandem with these research topics, this study presents a preliminary investigation of automatic identification of cover recordings, which tries to locate songs whose main melodies are similar to that of the fragment of the song specified by the user.

A cover version of a song refers to a new rendition of a song that was originally recorded and made popular by another artist. It is often used as a means to attract audiences who like a familiar song, or to increase the popularity of an artist by adapting a proven hit. Sometimes pop musicians gain publicity by recording a cover version that contrasts with the original recording. Over several years, thousands upon thousands of cover versions of songs have been recorded, some of which are virtually identical to the original version, while some are radically different. The only feature that is almost invariant in the different recordings is the main melody of the vocals. Usually, the main difference between an original song and a cover version is that they are performed by different singers. In such cases, the associated tempos, ornaments, accompaniments, *etc.*, may be changed to cater to the taste of contemporary audiences, or to fit the theme of an album. Thus, it would be useful if a music retrieval system incorporated a search function for a song rendered by different singers or belonging to different genres.

Other common differences between cover versions and the original song are that they have different lyrics and titles, or they are sung in different languages. Hit songs, in particular, are often translated into different languages, thereby making them more popular worldwide. Since a translation is usually not literal, cover-version identification based on the main melody could support a more feasible retrieval function than text-based retrieval for those wishing to listen to a song performed in a different language. In addition, it is commonplace for live performances to be recorded and then released as authorized cover songs. The method of cover-version identification could thus be applied to index and classify such undocumented live recordings. This would also help copyright holders detect unauthorized or bootleg concert recordings.

In this work, we address the problem of identifying cover-versions for music retrieval by investigating how to determine if one or more music recordings contain main melodies that are similar to a specified song excerpt. According to the categorization for music retrieval presented in Table 1¹, this task belongs to the category of retrieving polyphonic music documents based on *polyphonic* music queries. In contrast to textual or symbolic music retrieval, which can be handled by a number of conventional information-retrieval methods, the problems associated with acoustic documents/queries require digital audio signal processing, which involves many uncertain factors not considered by conventional information-retrieval methods. In addition, unlike monophonic music, in which only one note is played at any given time, polyphonic music often contains many notes that are played simultaneously. Consequently, it is difficult to extract the main melody from a piece of polyphonic music automatically [24]. Because of this difficulty, a large number of melody-based music retrieval systems [25-28] work within the monophonic domain, which converts a monophonic audio query into a symbolic format to match a monophonic symbolic collection. Some studies [29, 30] focus on locating the major themes from a piece of polyphonic symbolic music, in which the note information is given as *a priori*. However, very few systems operate in the mode of monophonic audio queries on a polyphonic audio collection [31, 32], or in the mode of entirely

¹ Summarized from [3, 22, 23].

Table 1. Problem categories in music-retrieval research. Depending on the combination of different types of music documents and users' queries, music-retrieval research could be divided into 25 problem categories. Example task for each category is given in the Table.

Query \ Document		Textual	Symbolic	Acoustic		Image
				Monophonic	Polyphonic	
Textual		<i>e.g.</i> , retrieving lyrics via keywords	<i>e.g.</i> , retrieving MIDI or Humdrum music via keywords	<i>e.g.</i> , retrieving solo trumpet music via keywords	<i>e.g.</i> , retrieving popular songs via keywords	<i>e.g.</i> , retrieving scanned sheet music via keywords
Symbolic		<i>e.g.</i> , retrieving lyrics via example MIDI music	<i>e.g.</i> , using one MIDI music to retrieve other MIDI versions	<i>e.g.</i> , retrieving solo trumpet music via example MIDI music	<i>e.g.</i> , retrieving popular songs via example MIDI music	<i>e.g.</i> , retrieving scanned sheet music via example MIDI music
Acoustic	Monophonic	<i>e.g.</i> , retrieving lyrics via humming	<i>e.g.</i> , retrieving MIDI music via humming	<i>e.g.</i> , retrieving solo trumpet music via humming	<i>e.g.</i> , retrieving popular songs via humming	<i>e.g.</i> , retrieving scanned sheet music via humming
	Polyphonic	<i>e.g.</i> , checking the source of a pre-recorded popular song	<i>e.g.</i> , retrieving MIDI versions of a pre-recorded popular song	<i>e.g.</i> , retrieving solo trumpet versions of a pre-recorded popular song	<i>e.g.</i> , retrieving original/cover versions of a popular song (the problem investigated in this study)	<i>e.g.</i> , retrieving scanned sheet music of a popular song
Image		<i>e.g.</i> , checking the source of a song via scanned sheet music	<i>e.g.</i> , retrieving MIDI music from scanned sheet music	<i>e.g.</i> , retrieving solo trumpet versions from scanned sheet music	<i>e.g.</i> , retrieving a symphony from scanned sheet music	<i>e.g.</i> , checking similar music via scanned sheet music

polyphonic audio queries on a polyphonic audio collection [33-35]. This work differs from the above systems because of the need to compare the main melody present in the vocals of polyphonic music. To tackle this problem, we propose methods for removing the non-vocal portions of a song, extracting the sung notes from the accompanied vocals, and comparing the similarities between the sung note sequences.

The remainder of the paper is organized as follows. An overview of the proposed method is given in section 2. The cover-version identification components, namely, non-vocal segment removal, main melody extraction, and similarity computation, are presented in sections 3, 4, and 5, respectively. We discuss the experiment results in section 6, and then present our conclusions in section 7.

2. METHOD OVERVIEW

Our goal is to design a system that takes an audio query from a fragment of a song as input, and produces, as output, a ranked list of songs that are similar to the query in terms of the main melody. Songs ranked high are then considered as either the original

version or cover versions of the song requested by the user. However, as cover versions may differ significantly from the original song in the way that the accompaniments are introduced, an arbitrary audio query could contain non-vocal (accompaniment-only) segments whose melody patterns are not present in the songs requested by the user, or vice versa. To simplify the problem during this initial development stage, we assume that a user's query does not contain salient non-vocal segments.

In general, the structure of a popular song can be divided into five sections: (1) *intro*, usually the first 5-20 seconds of the song, which is simply an instrumental statement of the subsequent sections; (2) *verse*, which typically comprises the main theme of the story represented in the song's lyrics; (3) *chorus*, which is often the heart of the song, where the most recognizable melody is present and repeated; (4) *bridge*, located roughly two-thirds into the song, where a key change, tempo change or new lyric is usually introduced to create a sensation of something new coming next; and (5) *outro*, which is often a fading version of the chorus or an instrumental restatement of some earlier sections to bring the song to a conclusion. Except for the intro and outro, the other sections may be repeated several times with different lyrics and melodies. For example, the song "Day Tripper" by *The Beatles* can be summarized as the structure "intro-verse-chorus-verse-chorus-bridge-verse-chorus-outro" [36]. In essence, the verse and chorus contain the vocals sung by the lead singer, while the intro, bridge, and outro are largely accompaniments. Since the vast majority of popular songs follow the structure of "intro-verse-chorus-verse-chorus-bridge-verse-chorus-outro", we further assume that a user's query is a fragment of the region between the intro and the bridge of a song.

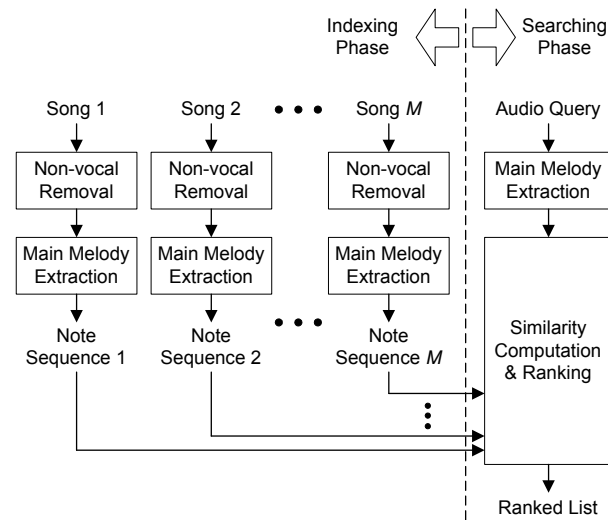


Fig. 1. The proposed cover-version identification system for music retrieval.

Fig. 1 shows a block diagram of our cover-version identification system for music retrieval, which operates in two phases: indexing and searching. The indexing phase generates the melody description for each of the songs (documents) in the collection. It commences by removing non-vocal segments longer than two seconds, which probably be-

long to the intro, bridge, or outro. Then, the main melody extraction component converts each song from waveform samples into a sequence of musical note symbols. In the search phase, the task is to determine which of the songs (documents) are relevant to a music query. This phase begins with main melody extraction, which converts an audio query into a sequence of musical note symbols, and is followed by comparison of the query's note sequence and each document's note sequence. The more similar the document's note sequence is to that of the query, the stronger the likelihood that the document is a cover version or the original version of the requested song. Then, a ranked list of the similarities between the query's sequence and the document's sequence is presented to the user.

3. NON-VOCAL SEGMENT REMOVAL

Although it would be advantageous if all the non-vocal regions in a music recording could be located automatically, the task of accurately distinguishing between segments with and without singing is rather difficult. In our previous work on this problem [37], we found that a vocal segment tends to be classified as non-vocal if it is mixed with loud background accompaniment. Although the effect of discarding a low “vocal-to-accompaniment-ratio” segment is almost negligible in some applications, such as singer clustering [37], it can result in a very fragmented and unnatural melody pattern being extracted from a song. Thus, instead of locating all the vocal and non-vocal boundaries in a song document, we only try to detect non-vocal segments that are longer than two seconds.

The basic strategy applied here is adapted from our previous work [37], in which a stochastic classifier is constructed to distinguish vocal from non-vocal regions. The classifier consists of a front-end signal processor that converts digital waveforms into frame-based cepstral feature vectors, after which a back-end statistical processor performs modeling and matching. The classifier operates in two phases, training and testing, as shown in Fig. 2.

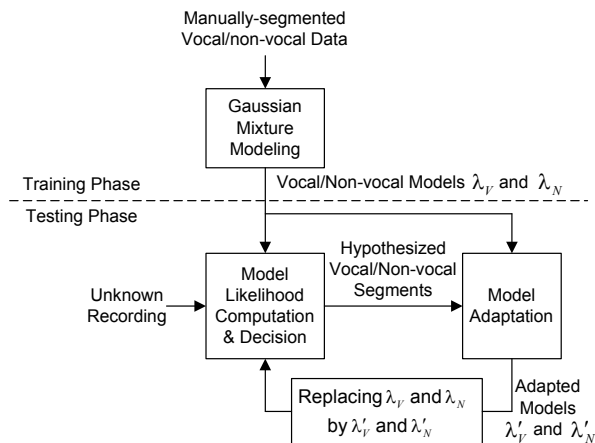


Fig. 2. Vocal/non-vocal classification.

During training, a music database with manual vocal/non-vocal transcriptions is used to form two separate Gaussian mixture models (GMMs): a vocal GMM, and a non-vocal GMM. Both GMMs are designed to model the spectral distribution of various broad acoustic classes by a combination of Gaussian components, in which the broad acoustic classes reflect some general vocal tract and instrumental configurations. We denote the vocal and non-vocal GMMs as λ_V and λ_N , respectively. The parameters of the GMMs are initialized via k -means clustering and iteratively adjusted via expectation-maximization (EM) [38].

In the testing phase, the recognizer takes as input the T_x -length feature vectors $X = \{x_1, x_2, \dots, x_{T_x}\}$ extracted from an unknown recording, and produces as output the frame log-likelihoods $\log p(x_t | \lambda_V)$ and $\log p(x_t | \lambda_N)$, for $1 \leq t \leq T_x$. Since singing tends to be continuous, it is better to perform the recognition task in a segment-by-segment manner, rather than a frame-by-frame manner. To reduce the risk of crossing multiple vocal/non-vocal boundaries, a segment is selected and examined in the following way. First, vector clustering is applied to all the frame feature vectors, and each frame is assigned a cluster index associated with its feature vector. Then, each segment is assigned the majority index of its constituent frames, and adjacent segments are merged as a homogeneous segment if they have the same index. Accordingly, a homogeneous segment can be hypothesized as either vocal or non-vocal by using the following equation:

$$\frac{1}{W_k} \left(\sum_{i=0}^{W_k-1} \log p(x_{s_k+i} | \lambda_V) - \sum_{i=0}^{W_k-1} \log p(x_{s_k+i} | \lambda_N) \right) \begin{matrix} \text{vocal} \\ > \\ \text{non-vocal} \end{matrix} \leq \eta, \quad (1)$$

where W_k and s_k represent, respectively, the length and the initial frame of the k th homogeneous segment, and η is the decision threshold.

As the performance of the above recognizer crucially depends on the reliability of the vocal/non-vocal models, it seems necessary to use training data that covers all the vocal/non-vocal characteristics of various music styles. However, acquiring such a large amount of training data is usually cost prohibitive because labeling the music manually requires considerable effort. To circumvent this problem, we propose tailoring the vocal/non-vocal models for each of the individual test music recordings, instead of designing models that can cover universal vocal/non-vocal characteristics. The idea is to refine the vocal/non-vocal models based on the recognition results. It is assumed that the acoustic characteristics of the true vocal/non-vocal segments in each music recording can be largely inferred from the hypothesized vocal/non-vocal segments. Thus, the hypothesized segments can be used to refine the models, and the recognizer with the refined models then repeats the procedure of likelihood computation and decision-making, which should improve recognition. There are a number of ways to refine the models. This study uses a model adaptation technique based on maximum *a posteriori* estimation [39]. The classification and model adaptation procedures are performed iteratively, until the resulting vocal/non-vocal boundaries do not change further. Finally, non-vocal segments longer than 2 seconds are identified and removed from the recording.

4. MAIN MELODY EXTRACTION

4.1 Note Sequence Generation

Given a music recording, the goal of main melody extraction is to find the sequence of musical notes produced by the singing part of the recording. Let $e_1, e_2, \dots, e_N$ be the inventory of possible notes performed by a singer. The task, therefore, is to determine which among N possible notes is most likely sung at each instant. To do this, the music signal is first divided into frames by using a fixed-length sliding Hamming window. Every frame then undergoes a fast Fourier transform (FFT) of size J . Since musical notes differ from each other in the fundamental frequencies (F0s) they represent, we can determine if a certain note is sung in each frame by analyzing the spectral intensity in the frequency region where the F0 of the note is located.

Let $x_{t,j}$ denote the signal's energy with respect to FFT index j in frame t , where $1 \leq j \leq J$. If we use the MIDI note number to represent $e_1, e_2, \dots, e_N$, and map the FFT indices into MIDI note numbers according to the F0 of each note, the signal's energy in note e_n of frame t can be estimated by

$$y_{t,n} = \max_{\forall j, U(j)=e_n} x_{t,j}, \quad (2)$$

and

$$U(j) = \left\lfloor 12 \cdot \log_2 \left(\frac{F(j)}{440} \right) + 69.5 \right\rfloor, \quad (3)$$

where $\lfloor \cdot \rfloor$ is a floor operator, $F(j)$ is the corresponding frequency of FFT index j , and $U(\cdot)$ represents a conversion between the FFT indices and the MIDI note numbers [28].

Ideally, if note e_n is sung in frame t , the resulting energy, $y_{t,n}$, should be the maximum among $y_{t,1}, y_{t,2}, \dots, y_{t,N}$. However, due to the existence of harmonics, note numbers that are several octaves higher than the sung note can also receive a large proportion of the signal's energy. Sometimes the energy of a harmonic note number can be even larger than the energy of the true sung note number; hence, the note number receiving the largest amount of energy is not necessarily the note that is sung. To determine the sung note more reliably, we adopt Sub-Harmonic Summation (SHS) [40] to solve the problem.

The principle applied here is to compute a value for the “strength” of each possible note by summing the signal's energy on a note and its harmonic note numbers. Specifically, the strength of note e_n in frame t is computed using

$$z_{t,n} = \sum_{c=0}^C h^c y_{t,n+12c}, \quad (4)$$

where C is the number of harmonics considered, and h is a positive value less than 1 that discounts the contribution of higher harmonics. The result of summation is that the note number corresponding to the signal's F0 receives the largest amount of energy from its harmonic notes. Thus, the sung note in frame t can be determined by choosing the note number associated with the largest value of the strength, *i.e.*,

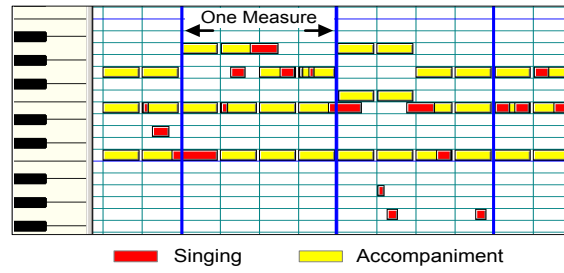


Fig. 3. An excerpt from the pop song “Let It Be” by *The Beatles* in which the tune is manually converted into a MIDI file.

$$o_t = \arg \max_{1 \leq n \leq N} z_{t,n}. \quad (5)$$

However, since most popular music contains background accompaniment during most or all vocal passages, the note number associated with the largest value of the strength may not be produced by a singer, but by the instruments being played concurrently instead. To alleviate the interference of the background accompaniment, we propose suppressing the strength of the notes that are probably produced by the instruments. The method is motivated by an observation made about popular music that, compared to the vocals, the principal accompaniment often contains a periodically-repeated note. Fig. 3 shows an example of a fragment of a pop song, where the tune is converted into a MIDI file. For ease of illustration, it is shown by the software CakewalkTM. From the figure, we observe that the melody produced by the principal accompaniment tends to be repeated in the adjacent measures, unlike the main melody produced by singing. Therefore, it can be assumed that a note number associated with the constantly-large value of the strength within and across adjacent measures is probably produced by the instruments. Based on this assumption, we modify the computation of strength in Eq. (4) by

$$\tilde{z}_{t,n} = z_{t,n} - \frac{1}{2(L_2 - L_1 + 1)} \left(\sum_{l=-L_2}^{-L_1} z_{t+l,n} + \sum_{l=L_1}^{L_2} z_{t+l,n} \right), \quad (6)$$

where L_1 and L_2 specify, respectively, the regions $[t - L_2, t - L_1]$ and $[t + L_1, t + L_2]$ in which the average strength of note e_n is computed. Implicit in Eq. (6) is that the strength of note e_n in frame t will be largely suppressed if the average strength of note e_n computed from the surrounding frames is large. Accordingly, the sung note in frame t is determined by

$$o_t = \arg \max_{1 \leq n \leq N} \tilde{z}_{t,n}. \quad (7)$$

4.2 Note Sequence Rectification

The above frame-based generation of note sequences may be improved by exploiting the underlying relation or constraints between frames. The most visible constraint between frames is that the length of a note is usually several times longer than a frame;

hence, there should not be a drastic change, such as jitter, between adjacent frames. To remove jitter from a note sequence, we apply median filtering, which replaces each note of the frame with the local median of its neighboring frames.

In addition to the short-term constraint between adjacent frames, we exploit a long-term constraint to rectify a note sequence. This constraint is based on the fact that the notes sung in a music recording usually vary far less than the range of all possible sung notes. Furthermore, the range of the notes sung within a verse or chorus section can be even narrower. Fig. 4 shows a segment of a pop song, in which the singing part is converted into a MIDI file. It is clear that the range of the notes in the verse can be distinguished from that of the chorus, mainly because the sung notes within a section are not spread over the range of all the possible notes, but are only distributed within their own narrower range. An informal survey using 50 pop songs shows that the range of sung notes in a complete song and in a verse or chorus section is around 24 and 22 semitones, respectively. Fig. 5 details our statistical results. The range of sung notes serves as a long-term constraint to rectify a note sequence.

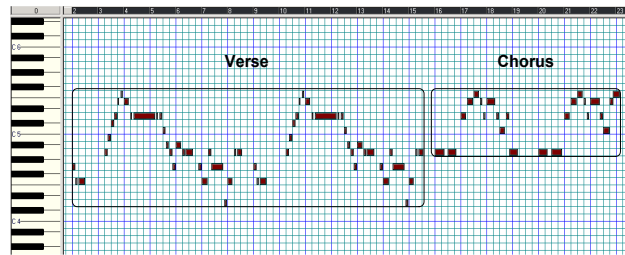
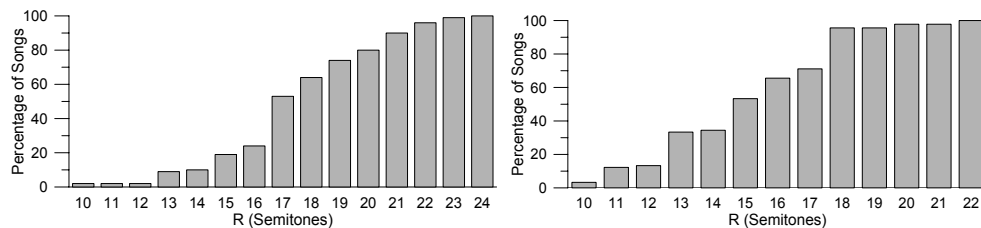


Fig. 4. An excerpt from the pop song “Yesterday” by *The Beatles*; the singing is converted into a MIDI file.



(a) The range of sung notes in a complete song. (b) The range of sung notes in a verse or chorus.

Fig. 5. Statistics of the range of sung notes in 50 pop songs, in which the percentage of songs whose range of sung notes is less than R semitones is shown.

The principle of rectification involves locating incorrectly estimated notes that result in a note sequence beyond the normal range. Since the accompaniment is often played several octaves above or below the vocals, the incorrectly estimated notes are probably the octaves of their true notes. Therefore, we can adjust some problematic notes by moving them several octaves up or down, so that the range of notes in an adjusted sequence conforms to the normal range. Specifically, let $o = \{o_1, o_2, \dots, o_T\}$ denote a note

sequence estimated using Eq. (7). Then, an adjusted note sequence $o' = \{o'_1, o'_2, \dots, o'_T\}$ is obtained by

$$o'_t = \begin{cases} o_t, & \text{if } |o_t - \bar{o}| \leq (R/2) \\ o_t - 12 \times \left\lfloor \frac{o_t - \bar{o} + R/2}{12} \right\rfloor, & \text{if } o_t - \bar{o} > (R/2) \\ o_t - 12 \times \left\lfloor \frac{o_t - \bar{o} - R/2}{12} \right\rfloor, & \text{if } o_t - \bar{o} < (-R/2) \end{cases}, \quad (8)$$

where R is the normal range of the sung notes in a sequence, say 24, and $\bar{o}$ is the mean note computed by averaging all the notes in o . In Eq. (8), a note, o_t , is considered incorrect and needs to be adjusted if it is too far from $\bar{o}$, *i.e.*, $|o_t - \bar{o}| > R/2$. The adjustment is performed by moving the incorrect note $\lfloor (o_t - \bar{o} + R/2)/12 \rfloor$ or $\lfloor (o_t - \bar{o} - R/2)/12 \rfloor$ octaves.

5. SIMILARITY COMPUTATION

After representing music data as a sequence of note numbers, cover-version identification can be transformed into a problem of comparing the similarity between a query's note sequence and each of the documents' note sequences. Since cover versions often differ from the original song in terms of key, tempo, ornament, *etc.*, it is virtually impossible to find a document sequence that matches the query sequence exactly. Moreover, main melody extraction is known to be frequently imperfect, which introduces errors of substitution, deletion, and insertion into the note sequences. To ensure reliable melody similarity comparison, an approximate matching method tolerant of occasional note errors is therefore needed.

Let $q = \{q_1, q_2, \dots, q_T\}$, and $u = \{u_1, u_2, \dots, u_L\}$ be, respectively, the note sequences extracted from a user's query and a music document to be compared. The obvious problem we face is that the lengths of q and u are usually unequal. Thus, it is necessary to temporally align q and u before computing their similarity. For this reason, we apply Dynamic Time Warping² (DTW) to find the mapping between each q_t and u_ℓ , $1 \leq t \leq T$, $1 \leq \ell \leq L$. DTW operates by constructing a $T \times L$ distance matrix $D = [D(t, \ell)]_{T \times L}$, where $D(t, \ell)$ is the distance between the note sequences $\{q_1, q_2, \dots, q_t\}$ and $\{u_1, u_2, \dots, u_\ell\}$. It is computed by

$$D(t, \ell) = \min \begin{cases} D(t-2, \ell-1) + 2 \times d(t, \ell) \\ D(t-1, \ell-1) + d(t, \ell) - \varepsilon \\ D(t-1, \ell-2) + d(t, \ell) \end{cases}, \quad (9)$$

and

$$d(t, \ell) = |q_t - u_\ell|, \quad (10)$$

where ε is a small constant that favors the mapping between notes q_t and u_ℓ , given the

² Similar work can be found in [27, 28, 41].

distance between note sequences $\{q_1, q_2, \dots, q_{t-1}\}$ and $\{u_1, u_2, \dots, u_{t-1}\}$. The boundary conditions for the above recursion are defined by

$$\begin{cases} D(1, 1) = d(1, 1) \\ D(t, 1) = \infty, \quad 2 \leq t \leq T \\ D(2, 2) = d(1, 1) + d(2, 2) - \varepsilon \\ D(2, 3) = d(1, 1) + d(2, 2) \\ D(3, 2) = d(1, 1) + 2 \times d(2, 2) \\ D(t, 2) = \infty, \quad 4 \leq t \leq T \end{cases} \quad (11)$$

After the distance matrix D has been constructed, the similarity between q and u can be evaluated by

$$S(q, u) = \begin{cases} \max_{T/2 \leq \ell \leq \min(2T, L)} [1/D(T, \ell)], & \text{if } L \geq T/2 \\ \infty, & \text{if } L < T/2 \end{cases} \quad (12)$$

Here, we assume that (1) the end of a query's sequence should be aligned to a certain frame between $T/2$ and $\min(2T, L)$ of the document's sequence; and (2) a document whose sequence length is less than $T/2$ is not relevant to the query.

Since a song query may be performed in a different key or register than the target music document, *i.e.*, the so-called *transposition*, the resulting note sequences of the query and the document could be rather different. To deal with this problem, the dynamic range of a query's note sequence needs to be adjusted to that of the document to be compared. This can be done by moving the query's note sequence up or down several semitones, so that the mean of the note sequence is equal to that of the document under consideration. A query's note sequence is adjusted by

$$q_t \leftarrow q_t + (\bar{u} - \bar{q}), \quad (13)$$

where $\bar{q}$ and $\bar{u}$ are the means of the query's note sequence and the document's note sequence, respectively. However, in our experiments, we found that the above adjustment could not fully resolve the transposition problem, since the value of $(\bar{q} - \bar{u})$ only reflected a global difference in key between a query and a document. In other words, the adjustment cannot characterize partial transpositions or key changes over the course of a query. To handle this problem better, we further modify the DTW similarity comparison by considering the key shifts of a query's note sequence. Specifically, a query sequence q is shifted by $\pm 1, \pm 2, \dots, \pm K$ semitones to span a set of note sequences $\{q^{(1)}, q^{(-1)}, q^{(2)}, q^{(-2)}, \dots, q^{(K)}, q^{(-K)}\}$. For a document sequence u , the similarity $S(q, u)$ is then determined by choosing the sequence among $\{q^{(0)}, q^{(1)}, q^{(-1)}, q^{(2)}, q^{(-2)}, \dots, q^{(K)}, q^{(-K)}\}$ that is most similar to u , *i.e.*,

$$S(q, u) = \max_{-K \leq k \leq K} S(q^{(k)}, u), \quad (14)$$

where $q^{(0)} = q$.

6. EXPERIMENTS

6.1 Music Data

The music database used in this study consisted of 794 tracks³ from pop music CDs that covered the following five genres: soundtracks, country, folk, jazz, and rock. We divided the database into three sub-sets. The first sub-set, DB-1, contained 47 pairs of tracks (a total of 94 tracks) comprised of cover/original songs. In this sub-set, the difference between a cover version and the original song was characterized by the following factors: L: language (including English, Mandarin, and Japanese); S: singer; A: principal accompaniment; T: tempo; and N: non-vocal melodies. A summary of the differences in each pair of tracks is given in Table 2.

Table 2. A summary of the differences in each pair of cover/original tracks in sub-set DB-1.

Type of within-pair difference	No. of pairs
L	8
L + S	7
L + T	3
L + S + T	7
L + T + N	6
L + S + T + N	4
L + A + T + N	2
L + S + A + T + N	10

The second sub-set, DB-2, contained 500 tracks, none of which was a cover version of any track in DB-1, but some of the singers in DB-2 also appeared in DB-1. The third sub-set, DB-3, contained 200 tracks, performed by 13 female and 8 male singers, none of whom appeared in DB-1 and DB-2. Sub-sets DB-1 and DB-2 were used to evaluate the cover-version retrieval system, while DB-3 was used to create the vocal and non-vocal models. Manual annotation of vocal/non-vocal boundaries was only performed on DB-1 and DB-3. To exclude high frequency components whose vocal information is usually sparse, the waveform signals were down-sampled from a CD sampling rate of 44.1 kHz to 22.05 kHz, to exclude high frequency components whose vocal information is usually sparse.

6.2 Experiment Results

The first experiment, using DB-1 as test data, was run in a “leave-one-out” manner, whereby one track at a time in DB-1 was used as a trial query to retrieve the remaining 93 tracks. We rotated through all 94 tracks in this manner. To approximate a real-use scenario, each query was composed of only a verse or chorus, obtained by manual segmentation. The length of the queries ranged from 31 to 54 seconds. The performance of the song retrieval method was evaluated in terms of the retrieval accuracy as follows:

³ The database did not contain the 50 pop songs described in section 4.2, which were used for analyzing the range of sung notes.

$$\frac{\# \text{ queries whose target songs are ranked first}}{\# \text{ queries}} \times 100\%.$$

We also computed the Top-N accuracy rate, defined as the percentage of queries whose target songs are among the Top-N.

Table 3. Performance of cover-version retrieval for different configurations used in main melody extraction; each method is based on five-frame median filtering.

Main melody extraction method		Accuracy (%)		
		Top 1	Top 3	Top 10
Eq. (5)		60.64	71.28	78.72
Eq. (7) ($L_1 = 64$, and $L_2 = 192$)		70.21	73.40	80.85
Eqs. (7) and (8)	$R = 22$	65.96	72.34	74.47
	$R = 24$	76.60	78.72	87.23
	$R = 26$	74.47	77.66	86.17
	$R = 28$	70.21	77.66	85.11

Table 3 shows the retrieval results for different configurations used in main melody extraction. In this experiment, each document was a track from which non-vocal segments had been removed manually. The inventory of possible sung notes consisted of the MIDI numbers from 41 to 83, which corresponded to the frequency range of 87 to 987 Hz. In FFT computation, the frame length and the overlap between frames were set to 2048 and 1704, respectively. In addition, for melody similarity comparison, we used $K = 2$ in Eq. (14) to handle the transposition problem. From Table 3, we observe that the retrieval performance obtained by using Eq. (5) was the least effective of the three methods compared. This is because the method determines the sung notes based on the strength computed from the observed signal, which is vulnerable to interference from the background accompaniment. It is clear from Table 3 that a better estimation of the note strength can be obtained by using Eq. (7), which discounts the note numbers associated with the constantly-large values of the strength within and across adjacent measures. Table 3 also shows that melody extraction can be further improved by using the note sequence rectification method defined in Eq. (8).

Table 4 shows the retrieval results for the different configurations used for melody similarity comparison. In this experiment, main melody extraction was performed using Eqs. (7) and (8) with $R = 24$, *i.e.*, the best results reported in Table 3. The results in Table 4 show that the retrieval performance improves as the value of K increases. This indicates that the more the possible changes of key are taken into account, the greater the chance that a query's sequence will match the correct document's sequence. However, increasing the value of K substantially increases the computational costs because similarity comparison requires two extra DTW operations whenever the value of K is increased by one. An economic value of $K = 2$ was thus chosen for all the experiments.

Next, we examined the performance of cover version retrieval based on the automatic removal of the non-vocal segments of each document. The number of Gaussian densities used in the vocal and non-vocal models was empirically determined to be 64;

Table 4. Performance of cover-version retrieval for different configurations used in melody similarity comparison.

Value of K in Eq. (14)	Accuracy (%)		
	Top 1	Top 3	Top 10
0	64.89	67.02	77.66
1	73.40	75.53	80.85
2	76.60	78.72	87.23
3	76.60	79.79	88.30

Table 5. Performance of cover-version retrieval obtained by removing and not removing the non-vocal segments of each document.

Non-vocal segment removal method	Accuracy (%)		
	Top 1	Top 3	Top 10
Manual removal	76.60	78.72	87.23
Automatic removal	65.96	69.15	72.34
Without removal	54.26	59.57	64.89

Table 6. Results of cover-version retrieval for the collection of 594 tracks in DB-1 and DB-2.

Non-vocal segment removal method	Accuracy (%)		
	Top 1	Top 3	Top 10
Automatic removal	63.83	65.96	72.34
Without removal	47.87	54.26	60.64

and the length of a segment, W , in Eq. (1) was set at 200. Table 5 shows the experiment results, in which the results of “Manual removal” correspond to the results of “ $K = 2$ ” in Table 4. From Table 5, we observe that, although there is a significant performance gap between the manual and automatic removal of the non-vocal segments, the performance obtained with automatic non-vocal removal is much better than that obtained without non-vocal removal.

We also conducted experiments to evaluate the retrieval performance of the proposed system for a larger collection of songs. We used the 94 queries, one at a time, to retrieve the 593 tracks in DB-1 and DB-2. Since no manual annotation of vocal/non-vocal boundaries was performed on DB-2, the experiment was run on the basis of automatic removal of the non-vocal segments of each document. Table 6 shows the experiment results. As expected, the increased number of non-target songs reduced the retrieval accuracy. Comparing Table 6 with Table 5, we find that the retrieval accuracy deteriorates sharply when the system operates on a larger collection of songs without removing the non-vocal segments. Once again, this demonstrates the importance of removing non-vocal regions.

Fig. 6 details the retrieval results for the 94 trial queries, where each point indicates the rank of each query’s target song among the 593 documents. Nearly all the target songs of queries belonging to “L” and “L + T” were ranked among the Top 3, whereas a large proportion of the target songs of queries belonging to “L + S + A + T + N” were ranked outside the Top 10. This reflects the fact that the greater the difference between

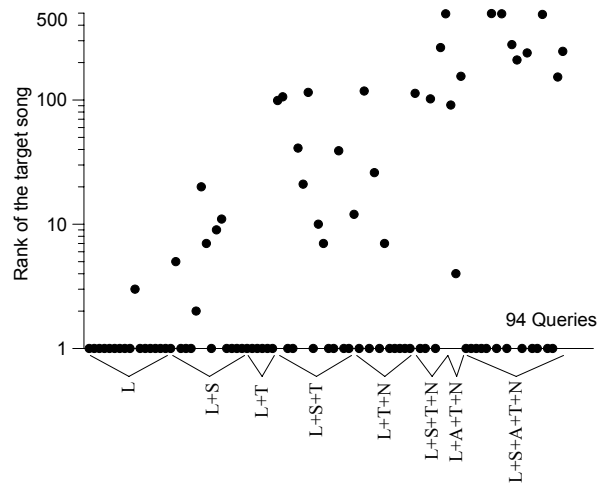


Fig. 6. The rankings of the 94 queries' target songs.

the cover version and the original song, the more difficult it is to retrieve one song by using another song as a query. Although the overall performance could be improved further, our system shows the feasibility of identifying polyphonic cover recordings for music retrieval.

7. CONCLUSION

We have examined the feasibility of identifying cover versions of popular songs for music retrieval. The proposed system tries to determine which songs in a database of songs contain similar main melodies to the melody in a user's query. To exclude factors that are irrelevant to the main melody of a song, we remove non-vocal segments that are longer than a whole rest. We have also proposed a method to minimize the interference of background accompaniment during the estimation of the sung note at each instant. The estimated sung note sequences are then further refined by limiting the range of sung notes in a sequence to 24 semitones. In addition, we have devised a method for comparing the similarity of a query's note sequence with the documents' note sequences. The method can handle the discrepancies in tempo and transposition between original songs and cover versions of them.

Despite the potential of the methods proposed in this study, they only provide baseline solutions to the cover-version retrieval problem. Like other research on retrieving polyphonic documents based on polyphonic queries, more work is needed to improve both melody extraction and melody similarity comparison. The improvement should not only be made on the system's effectiveness, but also be made on the efficiency. In particular, one of the factors that dominate our system's efficiency is the DTW-based similarity comparison, which may be too costly expensive to deal with a large number of music documents. To accelerate the similarity comparison, a strategy based on key-invariant encoding [42] could be incorporated into our system. In addition, the tradeoff between system's effectiveness and efficiency needs to be further studied. As a result, to

support the future work on cover-version retrieval, the music database should be scaled up to cover a wider variety of music styles, genres, singers, and languages.

REFERENCES

1. J. L. Hsu, C. C. Liu, and A. L. P. Chen, "Discovering nontrivial repeating patterns in music data," *IEEE Transactions on Multimedia*, Vol. 3, 2001, pp. 311-325.
2. W. Chai and B. Vercoe, "Music thumbnailing via structural analysis," in *Proceedings of the 11th ACM International Conference on Multimedia*, 2003, pp. 223-226.
3. C. Yang, "Efficient acoustic index for music retrieval with various degrees of similarity," in *Proceedings of the 10th ACM International Conference on Multimedia*, 2002, pp. 584-591.
4. M. A. Akeroyd, B. C. J. Moore, and G. A. Moore, "Melody recognition using three types of dichotic-pitch stimulus," *Journal of the Acoustical Society of America*, Vol. 110, 2001, pp. 1498-1504.
5. A. S. Durey and M. A. Clements, "Features for melody spotting using hidden Markov models," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2, 2002, pp. 1765-1768.
6. M. Goto and Y. Muraoka, "Real-time beat tracking for drumless audio signals: chord change detection for musical decisions," *Speech Communication*, Vol. 27, 1999, pp. 311-335.
7. M. Alonso, B. David, and G. Richard, "Tempo and beat estimation of musical signals," in *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004, pp. 158-163.
8. A. T. Cemgil, H. J. Kappen, and D. Barber, "A generative model for music transcription," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, 2006, pp. 679-694.
9. M. A. Akeroyd, B. C. J. Moore, and G. A. Moore, "Melody recognition using three types of dichotic-pitch stimulus," *Journal of the Acoustical Society of America*, Vol. 110, 2001, pp. 1498-1504.
10. G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, Vol. 10, 2002, pp. 293-302.
11. T. Li, M. Ogihara, and Q. Li, "A comparative study on content-based music genre classification," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, pp. 282-289.
12. Y. E. Kim and B. Whitman, "Singer identification in popular music recordings using voice coding features," in *Proceedings of the 3rd International Conference on Music Information Retrieval*, 2002, pp. 164-169.
13. W. H. Tsai and H. M. Wang, "Automatic singer recognition of popular music recordings via estimation and modeling of solo vocal signals," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 14, 2006, pp. 330-341.
14. P. Herrera, X. Amatriain, E. Batlle, and X. Serra, "Towards instrument segmentation for music content description: a critical review of instrument classification techniques," in *Proceedings of International Symposium on Music Information Retrieval*, 2000, http://ciir.cs.umass.edu/music2000/papers/herrera_paper.pdf.
15. A. Eronen, "Musical instrument recognition using ICA-based transform of features

- and discriminatively trained HMMS,” in *Proceedings of the 7th International Symposium on Signal Processing and its Applications*, Vol. 2, 2003, pp. 133-136.
16. J. Haitisma and T. Kalker, “A highly robust audio fingerprinting system,” in *Proceedings of International Conference on Music Information Retrieval*, 2002, pp. 144-148.
 17. V. Venkatachalam, L. Cazzanti, N. Dhillon, and M. Wells, “Automatic identification of sound recordings,” *IEEE Signal Processing Magazine*, Vol. 21, 2004, pp. 92-99.
 18. W. Birmingham, R. Dannenberg, and B. Pardo, “Query by humming with the vocal-search system,” *Communications of the ACM*, Vol. 49, 2006, pp. 49-52.
 19. J. W. Dunn, D. Byrd, M. Notess, J. Riley, and R. Scherle, “Variations2: retrieving and using music in an academic setting,” *Communications of the ACM*, Vol. 49, 2006, pp. 53-58.
 20. A. Ferrara, L. A. Ludovico, S. Montanelli, S. Castano, and G. Haus, “A semantic web ontology for context-based classification and retrieval of music resources,” *ACM Transactions on Multimedia Computing, Communications and Applications*, Vol. 2, 2006, pp. 177-198.
 21. J. Foote, “An overview of audio information retrieval,” *ACM Multimedia System*, Vol. 7, 1999, pp. 42-51.
 22. N. Orio, “Music retrieval: a tutorial and review,” *Foundations and Trends in Information Retrieval*, Vol. 1, 2006, pp. 1-90.
 23. http://www.music-ir.org/research_home.html.
 24. J. Eggink and G. J. Brown, “Extracting melody lines from complex audio,” in *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004, pp. 84-91.
 25. A. Ghias, H. Logan, D. Chamberlin, and B. C. Smith, “Query by humming: musical information retrieval in an audio database,” in *Proceedings of the ACM International Conference on Multimedia*, 1995, pp. 231-236.
 26. N. Kosugi, T. Nishihara, S. Sakata, M. Yamamuro, and K. Kushima, “A practical query-by-humming system for a large music database,” in *Proceedings of the 8th ACM Conference on Multimedia*, 2000, pp. 333-342.
 27. N. Hu and R. B. Dannenberg, “A comparison of melodic database retrieval techniques using sung queries,” in *Proceedings of the 2nd ACM/IEEE-CS Joint Conference on Digital Libraries*, 2002, pp. 301-307.
 28. S. Pauws, “CubyHum: a fully operational query by humming system,” in *Proceedings of the 3rd International Conference on Music Information Retrieval*, 2002, pp. 187-196.
 29. C. Meek and W. P. Birmingham, “Automatic thematic extractor,” *Journal of Intelligent Information Systems*, Vol. 21, 2003, pp. 9-33.
 30. R. Typke, R. C. Veltkamp, and F. Wiering, “Searching notated polyphonic music using transportation distances,” in *Proceedings of the 12th Annual ACM Conference on Multimedia*, 2004, pp. 128-135.
 31. 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 *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval*, 2001, pp. 211-218.
 32. J. Song, S. Y. Bae, and K. Yoon, “Mid-level music melody representation of poly-

- phonic audio for query-by-humming system,” in *Proceedings of the 1st Annual International Conference on Music Information Retrieval*, 2002, pp. 133-139.
33. S. Doraisamy and S. M. Ruger, “An approach towards a polyphonic music retrieval system,” in *Proceedings of the 2nd Annual International Symposium on Music Information Retrieval*, 2001, pp. 187-193.
 34. J. Pickens, J. P. Bello, G. Monti, T. Crawford, M. Dovey, M. Sandler, and D. Byrd, “Polyphonic score retrieval using polyphonic audio queries: a harmonic modelling approach,” in *Proceedings of the 3rd Annual International Conference on Music Information Retrieval*, 2002, pp. 140-149.
 35. J. Foote, “ARTHUR: retrieving orchestral music by long-term structure,” in *Proceedings of the 1st Annual International Symposium on Music Information Retrieval*, 2000, http://ciir.cs.umass.edu/music2000/papers/foote_paper.pdf.
 36. S. Pauws, “Effects of song familiarity, singing training and recent song exposure on the singing of melodies,” in *Proceedings of the 4th International Conference on Music Information Retrieval*, 2003, pp. 57-64.
 37. W. H. Tsai, H. M. Wang, and D. Rodgers, “Blind clustering of popular music recordings based on singer voice characteristics,” *Computer Music Journal*, Vol. 28, 2004, pp. 68-78.
 38. A. Dempster, N. Laird, and D. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal Royal Statistical Society*, Vol. 39, 1977, pp. 1-38.
 39. D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital Signal Processing*, Vol. 10, 2000, pp. 19-41.
 40. M. Piszczalski and B. A. Galler, “Predicting musical pitch from component frequency ratios,” *Journal of the Acoustical Society of America*, Vol. 66, 1979, pp. 710-720.
 41. B. Pardo, W. P. Birmingham, and J. Shifrin, “Name that tune: a pilot study in finding a melody from a sung query,” *Journal of the American Society for Information Science and Technology*, Vol. 55, 2004, pp. 283-300.
 42. Y. H. Tseng, “Content-based retrieval for music collections,” in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 176-182.



Wei-Ho Tsai (蔡偉和) received his B.S. degree in Electrical Engineering from National Sun Yat-Sen University, Kaohsiung, Taiwan, R.O.C., in 1995. He received his M.S. and Ph.D. degrees in Communication Engineering from National Chiao-Tung University, Hsinchu, Taiwan, in 1997 and 2001, respectively. From 2001 to 2003, he was with Philips Research East Asia, Taipei, Taiwan, where he worked on speech processing problems in embedded systems. From 2003 to 2005, he served as a Postdoctoral Fellow at the Institute of Information Science, Academia Sinica, Taipei, Taiwan. He is currently an Assistant Professor in the Department of Electronic Engineering and Graduate Institute of Computer and Communication Engineering, National Taipei University of Technology, Taiwan. His research interests include spoken language processing and music information retrieval.



Hung-Ming Yu (游弘明) received his B.S. and M.S. degrees in Computer Science and Information Engineering from Chung Hua University, Hsinchu, Taiwan, R.O.C., in 2000 and 2002, respectively. He joined the Institute of Information Science of Academia Sinica in 2002 as a research assistant. His research interests include music information retrieval, multimedia databases, and content-based retrieval.



Hsin-Min Wang (王新民) received his B.S. and Ph.D. degrees in Electrical Engineering from National Taiwan University in 1989 and 1995, respectively. In October 1995, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, R.O.C., as a Postdoctoral Fellow. He was promoted to Assistant Research Fellow and then Associate Research Fellow in 1996 and 2002, respectively. He used to be an adjunct Associate Professor at National Taipei University of Technology and National Chengchi University. His major research interests include speech processing, natural language processing, spoken dialogue processing, multimedia information retrieval, and pattern recognition. He currently serves on the editorial board of the International Journal of Computational Linguistics and Chinese Language Processing. He was a recipient of the Chinese Institute of Engineers (CIE) Technical Paper Award in 1995. He is a senior member of IEEE, a life member of ACLCLP and IICM, and a member of ISCA.