LyricSynchronizer: Automatic Synchronization System Between Musical Audio Signals and Lyrics

Hiromasa Fujihara, Masataka Goto, Jun Ogata, and Hiroshi G. Okuno

***Abstract*—This paper describes a system that can automatically synchronize polyphonic musical audio signals with their corresponding lyrics. Although methods for synchronizing monophonic speech signals and corresponding text transcriptions by using Viterbi alignment techniques have been proposed, these methods cannot be applied to vocals in CD recordings because vocals are often overlapped by accompaniment sounds. In addition to a conventional method for reducing the influence of the accompaniment sounds, we therefore developed four methods to overcome this problem: a method for detecting vocal sections, a method for constructing robust phoneme networks, a method for detecting fricative sounds, and a method for adapting a speech-recognizer phone model to segregated vocal signals. We then report experimental results for each of these methods and also describe our music playback interface that utilizes our system for synchronizing music and lyrics.**

***Index Terms*—Alignment, lyrics, singing voice, Viterbi algorithm, vocal.**

I. Introduction

Since the lyrics of a song represent its theme and story, they are essential to creating an impression of the song. This is why music videos often help the audience enjoy the music by displaying synchronized lyrics as a caption. When a song is heard, for example, some people listen to the vocal melody and follow the lyrics.

In this paper, we describe a system that synchronizes the polyphonic audio signals and the lyrics of songs automatically by estimating the temporal relationship (alignment) between the audio signals and the corresponding lyrics. This approach is different from direct lyrics recognition and takes advantage of the vast selections of lyrics available on the web. Our system has a number of applications, such as automatic generation of music video captions and a music playback interface that can directly access to specific words or passages of interest.

Wang *et al.* developed a system called LyricAlly [1] for synchronizing lyrics with music recordings without extracting singing voices from polyphonic sound mixtures. It uses the duration of each phoneme as a cue for synchronization but it is not always effective because phoneme duration varies and can be altered by musical factors such as the location in a melody. Wong *et al.* [2] developed an automatic synchronization system for Cantonese popular music. It uses the tonal characteristics of Cantonese language and compares the tone of each word in the lyrics with the fundamental frequency (F0) of the singing voice, but because most languages do not have the tonal characteristics of Cantonese, this system cannot be generalized to most other languages. Loscos *et al.* [3] used a speech recognizer for aligning a singing voice and Wang *et al.* [4] used a speech recognizer for recognizing a singing voice, but they assumed pure monophonic singing without accompaniment. Gruhne *et al.* [5] worked on phoneme recognition in polyphonic music. Assuming that boundaries between phonemes were given, they compared several classification techniques. Their experiments were preliminary, and there were difficulties in actually recognizing the lyrics.

Since current speech recognition techniques are incapable of automatically synchronizing lyrics with music that includes accompaniment, we used an accompaniment sound reduction method [6] as well as the following four methods: a method for detecting vocal sections, a method for detecting fricative sounds, a method for constructing a phoneme network that is robust to utterances not in the lyrics, and a method for adapting a phone model for speech to segregated vocal signals.

II. System for Automatically Synchronizing Music and Lyrics

Given musical audio signals and the corresponding lyrics, our system calculates the start and end times for each phoneme of the lyrics. The target data are real-world musical audio signals such as popular music CD recordings that contain a singer's vocal track and accompaniment sounds. We make no assumptions about the number and kind of sound sources in the accompaniment sounds. We assume that the main vocal part is sung by a single singer (except for choruses).

Because the ordinary Viterbi alignment (forced alignment) method used in automatic speech recognition is negatively influenced by accompaniment sounds performed together with a vocal and also by interlude sections in which the vocal is not performed, we first obtain the waveform of the melody by extracting and resynthesizing the harmonic structure of the melody using the accompaniment sound reduction method proposed in [6]. We then detect the vocal region in the separated melody's audio signal, using a *vocal activity detection* method based on a hidden Markov model (HMM). We also detect the fricative

Manuscript received September 30, 2010; revised March 01, 2011; accepted May 18, 2011. Date of publication June 16, 2011; date of current version September 16, 2011. This work was supported in part by CrestMuse, CREST, JST. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Daniel Ellis.

H. Fujihara, M. Goto, and J. Ogata are with the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba 305-8568, Japan (e-mail: h.fujihara@aist.go.jp; m.goto@aist.go.jp; jun.ogata@aist.go.jp).

H. G. Okuno is with Kyoto University, Kyoto 606-8501, Japan (e-mail: okuno@i.kyoto-u.ac.jp).

Digital Object Identifier 10.1109/JSTSP.2011.2159577

1932-4553/$26.00 © 2011 IEEE

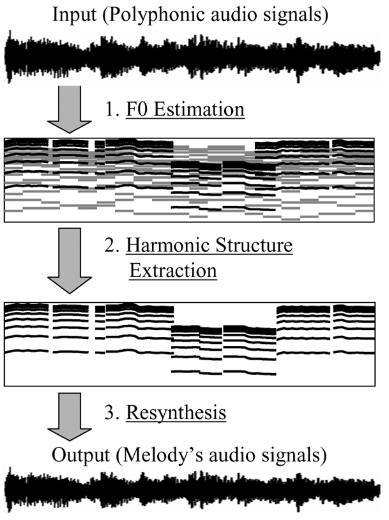
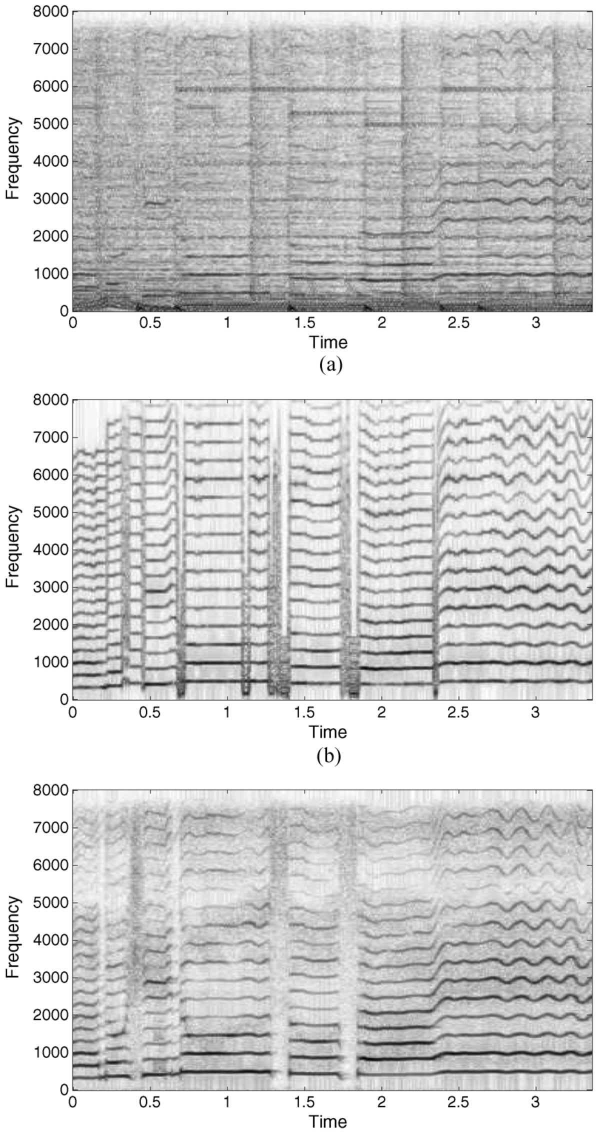

Input (Polyphonic audio signals)
1. F0 Estimation
2. Harmonic Structure Extraction
3. Resynthesis
Output (Melody's audio signals)

Fig. 1. Overview of accompaniment sound reduction.



(a)

(b)

(c)

Fig. 2. Example of accompaniment sound reduction taken from [6]. (a) A spectrogram of polyphonic signals. (b) A spectrogram of segregated signals. (c) A spectrogram of vocal-only signals.

sound by using a *fricative sound detection* method and incorporate this information into the next alignment stage. Finally, we align the lyrics and the separated vocal audio signals by using a *Viterbi alignment* method. The language model used in this alignment stage incorporates a *filler model* so that the system becomes robust to inter-phrase vowel utterances not written in the lyrics. We also propose a method for adapting a phone model to the separated vocal signals of the specific singer.

A. Accompaniment Sound Reduction

To extract a feature that represents the phonetic information of a singing voice from polyphonic audio signals, we need to reduce the accompaniment sound, as shown in Fig. 1. We do this by using a melody resynthesis technique based on a harmonic structure [6] consisting of the following three parts:

1) estimate the fundamental frequency (F0) of the melody by using Goto's PreFEst [7];
2) extract the harmonic structure corresponding to the melody;
3) resynthesize the audio signal (waveform) corresponding to the melody by using a sinusoidal synthesis.

We thus obtain a waveform corresponding only to the melody. Fig. 2 shows spectrograms of polyphonic musical audio signals, that of the audio signals segregated by the accompaniment sound reduction method, and that of the original (ground-truth) vocal-only signals. It can be seen that the harmonic structure of a singing voice is enhanced by using the accompaniment sound reduction method.

Note that the melody obtained this way contains instrumental (i.e., nonvocal) sounds in interlude sections as well as voices in vocal sections, because the melody is defined as merely the most predominant note in each frame [7]. Since long nonvocal sections negatively influence the execution of the Viterbi alignment between the audio signal and the lyrics, we need to remove the interlude sections. Vocal sections are therefore detected by using the method described in Section II-B. Furthermore, since this method is based on the harmonic structure of the singing voice, unvoiced consonants, which do not have harmonic structures, cannot be separated properly. We try to partially overcome this issue by using the fricative sound detection method described in Section II-C.

Since the accompaniment sound reduction method is executed as a preprocessing of feature extraction for Viterbi alignment, it is easy to replace this with other singing voice separation or an F0 estimation method [8]. In this paper, we adopt the PreFEst-based accompaniment sound reduction method because it was reported that PreFEst achieved higher performance in F0 estimation experiments of polyphonic singing voices [9].

1) F0 Estimation: We used Goto's PreFEst [7] to estimate the F0 of the melody line. PreFEst can estimate the most predominant F0 in frequency-range-limited sound mixtures. Since the melody line tends to have the most predominant harmonic structure in middle- and high-frequency regions, we can estimate the F0 of the melody line by applying PreFEst with adequate frequency-range limitations.

2) Harmonic Structure Extraction: By using the estimated F0, we then extract the amplitude of the fundamental frequency

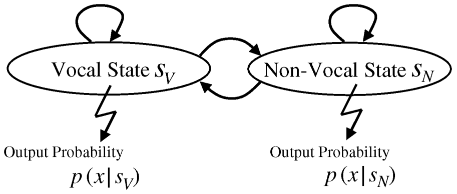


Fig. 3. Hidden Markov model (HMM) for vocal activity detection.

component and harmonic components. For each component, we allow r cent[1] error and extract the local maximum amplitude in the allowed area. The frequency $F_l^{(t)}$ and amplitude $A_l^{(t)}$ of the lth overtone $(l = 1, \ldots, L)$ at time (t) can be represented as

$$F_l^{(t)} = \underset{F}{\arg\max} \left|S^{(t)}(F)\right| \left(l\overline{F}^{(t)}\left(1-2^{\frac{r}{1200}}\right) \le F \le l\overline{F}^{(t)}\left(1+2^{\frac{r}{1200}}\right)\right) \tag{1}$$

$$A_l^{(t)} = \left|S^{(t)}(F_l)\right| \tag{2}$$

where $S^{(t)}(F)$ denotes the complex spectrum, and $\overline{F}^{(t)}$ denotes F0 estimated by the PreFEst. In our experiments, we set r to 20.

3) Resynthesis: Finally, we use a sinusoidal model to resynthesize the audio signal of the melody by using the extracted harmonic structure, $F_l^{(t)}$ and $A_l^{(t)}$. Changes in phase are approximated using a quadratic function so that the frequency can change linearly. Changes in amplitude are also approximated using a linear function.

B. Vocal Activity Detection

We propose a vocal activity detection method that can control the balance between the hit and correct rejection rates. There is generally a tradeoff relationship between the hit and correct rejection rates, and a proper balance between them depends on the application. For example, since our system positions the vocal activity detection method before the Viterbi alignment, the hit rate is more important than the probability of correct rejection because we want to detect all the regions that contain vocals. No previous studies on vocal activity detection [10]–[12] ever tried to control the balance between the probabilities.

1) Basic Formulation: We introduce a hidden Markov model (HMM) that transitions back and forth between vocal state, s_V, and non-vocal state, s_N, as shown in Fig. 3. Vocal state means that vocals are present and non-vocal state means that vocals are absent. Given the feature vectors of input audio signals, the problem is finding the most likely sequence of vocal and non-vocal states, $\hat{S} = \{s_1, \cdots, s_t, \cdots\}$:

$$\hat{S} = \underset{S}{\arg\max} \sum_t \{\log p(\boldsymbol{x}|s_t) + \log p(s_{t+1}|s_t)\} \tag{3}$$

where $p(\boldsymbol{x}|s)$ represents an output probability of state s, and $p(s_{t+1}|s_t)$ represents a state transition probability for the transition from state s_t to state s_{t+1}.

The output log probability of each state is approximated with the following equations:

$$\log p(\boldsymbol{x}|s_V) = \log \mathcal{N}_{\mathrm{GMM}}(\boldsymbol{x};\theta_V) - \frac{1}{2}\eta \tag{4}$$

$$\log p(\boldsymbol{x}|s_N) = \log \mathcal{N}_{\mathrm{GMM}}(\boldsymbol{x};\theta_N) + \frac{1}{2}\eta \tag{5}$$

where $\mathcal{N}_{\mathrm{GMM}}(\boldsymbol{x};\theta)$ denotes the probability density function of the Gaussian mixture model (GMM) with parameter θ, and η represents a threshold parameter that controls tradoff between the hit and correct rejection rates. The parameters of the vocal GMM, θ_{V}, and the nonvocal GMM, θ_{N}, are trained on feature vectors extracted from vocal sections and nonvocal sections of the training data set, respectively. We set the number of GMM mixtures to 64.

2) Calculation of Threshold: The balance of vocal activity detection is controlled by changing η in (4) and (5), but there is bias in the log likelihoods of the GMMs for each song, and it is difficult to decide the universal value of η. We therefore divide η into a bias correction value, $\eta_{\mathrm{dyn.}}$, and an application-dependent value, η_{fixed}:

$$\eta = \eta_{\mathrm{dyn.}} + \eta_{\mathrm{fixed}} \tag{6}$$

The bias correction value, $\eta_{\mathrm{dyn.}}$, is obtained from input audio signals by using Otsu's method for threshold selection [13] based on discriminant analysis, and the application-dependent value, η_{fixed}, is set by hand.

We first calculate the difference of log likelihood, $l(\boldsymbol{x})$, for all the feature vectors in input audio signals:

$$l(\boldsymbol{x}) = \log \mathcal{N}_{\mathrm{GMM}}(\boldsymbol{x};\theta_V) - \log \mathcal{N}_{\mathrm{GMM}}(\boldsymbol{x};\theta_N). \tag{7}$$

We then calculate the bias correction value, $\eta_{\mathrm{dyn.}}$, by using Otsu's method. The Otsu's method assumes that a set of $l(\boldsymbol{x})$ contains two classes of values and calculates the optimum threshold that maximizes their inter-class variance. When a histogram of $l(\boldsymbol{x})$ is denoted as $h(l)$, the inter-class variance $\sigma^2(\eta_{\mathrm{dyn.}})$ can be written as

$$\sigma^2(\eta_{\mathrm{dyn.}}) = \frac{\{\mu\omega(\eta_{\mathrm{dyn.}}) - \mu(\eta_{\mathrm{dyn.}})\}^2}{\omega(\eta_{\mathrm{dyn.}})\left(1-\omega(\eta_{\mathrm{dyn.}})\right)} \tag{8}$$

$$\omega(\eta_{\mathrm{dyn.}}) = \frac{\int_{\eta_{\mathrm{dyn.}}}^{\infty} h(l)dl}{\int_{-\infty}^{\infty} h(l)dl} \tag{9}$$

$$\mu(\eta_{\mathrm{dyn.}}) = \frac{\int_{\eta_{\mathrm{dyn.}}}^{\infty} lh(l)dl}{\int_{\eta_{\mathrm{dyn.}}}^{\infty} h(l)dl} \tag{10}$$

$$\mu = \frac{\int_{-\infty}^{\infty} lh(l)dl}{\int_{-\infty}^{\infty} h(l)dl}. \tag{11}$$

In practice, the threshold, $\eta_{\mathrm{dyn.}}$, can take only a finite number of values since the histogram, $h(l)$, is a discrete function. Thus, it is possible to calculate $\sigma^2(\eta_{\mathrm{dyn.}})$ for all possible $\eta_{\mathrm{dyn.}}$ and obtain the optimum value.

3) Novel Feature Vectors for Vocal Activity Detection: The vocal activity detection after the accompaniment sound reduction can be interpreted as a problem of judging whether the sound source of the given harmonic structure is vocal or nonvocal. In our previous system [6], we estimated the spectral envelope of the harmonic structure and evaluate the distance between it and the spectral envelopes in the training database.

[1] The cent is a logarithmic scale used for musical intervals in which the octave is divided into 1200 cents.

However, spectral envelopes estimated from high-pitched sounds by using cepstrum or linear prediction coding (LPC) analysis are strongly affected by spectral valleys between adjacent harmonic components. Thus, there are some songs (especially those sung by female singers) for which the vocal activity detection method did not work well.

This problem boils down to the fact that a spectral envelope estimated from a harmonic structure is not reliable except for the points (peaks) around each harmonic component. This is because a harmonic structure could correspond to different spectral envelopes: the mapping from a harmonic structure to its original spectral envelopes is a one-to-many association. When we consider this issue using sampling theory, the harmonic components are points sampled from their original spectral envelope at the interval of F0 along the frequency axis. The perfect reconstruction of the spectral envelope from the harmonic components is therefore difficult in general. Because conventional methods, such as Mel-frequency cepstral coefficient (MFCC) and LPC, estimate only one possible spectral envelope, the distance between two sets of the harmonic structure from the same spectral envelope is sometimes inaccurate. Though several studies have been proposed that have tried to overcome such instability of cepstrum [14], [15] by interpolating harmonic peaks or introducing new distance measures, such studies still have been trying to estimate a spectral envelope from an unreliable portion of the spectrum.

To overcome this problem, the distance must be calculated using only the reliable (sampled) points at the harmonic components. We focus on the fact that we can directly compare the power of harmonic components between two sets of the harmonic structure if their F0s are approximately the same. Our approach is to use the power of harmonic components directly as feature vectors and compare the given harmonic structure with only those in the database that have similar F0 values. This approach is robust against high-pitched sounds, because the spectral envelope does not need to be estimated. The powers of first to 20th overtones from the polyphonic audio signals are extracted and used as a feature vector.

To ensure that comparisons are made only with feature vectors that have similar F0s, we also use the F0 value as a feature in addition to the power of harmonic components. By using GMMs to model the feature vectors, we can be sure that each Gaussian can cover feature vectors that have similar F0s. When we calculate the likelihood of a GMM, the weights of the Gaussians that have large F0 values are minuscule. Thus, we can calculate the distance only with harmonic structures that have similar F0 values. There have been studies that used similar features in the field of sound source recognition [16]. These studies concern instrumental sounds, and it is not derived from an aspect of spectral envelope estimation.

The absolute value of the power of the harmonic structure is biased depending on the volume of each song. We therefore normalize the power of all harmonic components for each song. The normalized power of the hth harmonic component at time t, $p_h'^t$, is given by

$$p_h'^t = \exp\left(\log p_h^t - \frac{\sum_t \sum_h \log p_h^t}{T \times H}\right) \qquad (12)$$

where p_h^t represents the original power, T is the total number of frames, and H is the number of harmonic components considered. In this equation, an average power of every frequency bin of all the frames is subtracted from the original power in a log domain.

C. Use of Unvoiced Consonants Based on Fricative Detection

The forced alignment algorithm used in automatic speech recognition (ASR) synchronizes speech signals and texts by making phoneme networks that consist of all the vowels and consonants. However, since the accompaniment sound reduction, which is based on the harmonic structure of the melody, cannot segregate unvoiced consonants that do not have harmonic structure, it is difficult for the general forced alignment algorithm to align unvoiced consonants correctly unless we introduce a method for detecting unvoiced consonants from the original audio signals. We therefore developed a signal processing technique for detecting candidate unvoiced fricative sounds (a type of unvoiced consonant) in the input audio signals. Here, we focus on the unvoiced fricative sounds because their durations are generally longer than those of the other unvoiced consonants and because they expose salient frequency components in the spectrum.

1) Nonexistence Region Detection: It is difficult to accurately detect the existence of each fricative sound because the acoustic characteristics of some instruments (cymbals and snare drums, for example) sometimes resemble those of fricative sounds. If we take an approach such that we align /SH/ phoneme to frames if and only if they were detected as fricative regions, detection errors (no matter if they are false positive or false negative) can degrade the accuracy significantly in the later forced alignment step. We therefore take the opposite approach and try to detect regions in which there are no fricative sounds, i.e., *nonexistence* regions. Then, in the forced alignment, fricative consonants are prohibited from appearing in the nonexistence regions. However, if the frames including the /SH/ sound are erroneously judged as the nonexistence region, this kind of error affects the performance even in this approach; we can ameliorate this influence by setting a strict threshold and having a fricative detector to detect fewer regions as nonexistence regions.

2) Fricative Sound Detection: Fig. 4 shows an example spectrogram depicting non-periodic source components such as snare drum, fricative, and high-hat cymbal sounds in popular music. The characteristics of these non-periodic source components are depicted as vertical lines or clouds along the frequency axis in the spectrogram, whereas periodic source components tend to have horizontal lines. In the frequency spectrum at a certain time, these vertical and horizontal lines, respectively, correspond to flat and peaked (pointed) components.

To detect flat components from non-periodic sources, we need to ignore peak components in the spectrum. We therefore use the bottom envelope estimation method proposed by Kameoka *et al.* [17]. As shown in Fig. 5, the bottom envelope is defined as the envelope curve that passes through spectral valleys. The function class of the bottom envelope is defined as

$$g(f, \boldsymbol{a}) = \sum_{i=1}^{I} a_i \mathcal{N}(f; 400 \times i, 200^2) \qquad (13)$$

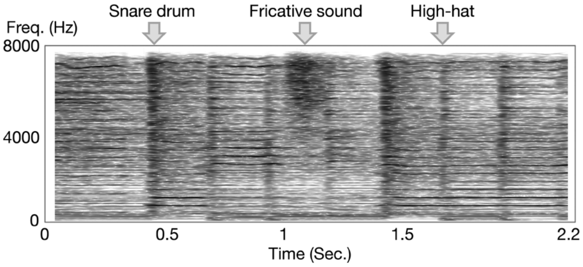


Fig. 4. Example spectrogram depicting snare drum, fricative, and high-hat cymbal sounds.

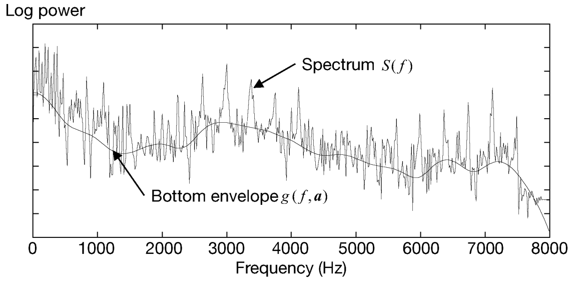


Fig. 5. Bottom envelope $g(f, \boldsymbol{a})$ in a spectrum $S(f)$.

where f denotes the frequency in Hz, $\mathcal{N}(x; m, \sigma^2)$ is the Gaussian function, and $\boldsymbol{a} = (a_1, \cdots, a_I)$ represents the weights of each Gaussian. This function class approximates arbitral spectral envelopes by using the weighted sum of Gaussian functions of which the means and variances are fixed. The means of the Gaussians are set so that they equally align to the frequency axis, and their variances are set so that the shape of this function class becomes smooth.

The problem here is to estimate $\boldsymbol{a}$, which determines the envelope curve. We therefore estimate the $\hat{\boldsymbol{a}}$ that minimizes the objective function

$$J = \int \left(\frac{g(f; \boldsymbol{a})}{S(f)} - \log \frac{g(f; \boldsymbol{a})}{S(f)} \right) df \tag{14}$$

where $S(f)$ represents the spectrum at each frame. This objective is derived by reversing $g(f; \boldsymbol{a})$ and $S(f)$ in the Itakura–Saito distance. Unlike the Itakura–Saito distance that penalizes positive errors much more than negative ones and is used to estimate the top envelope of a spectrum, this objective function penalizes negative errors much more than positive ones to estimate the bottom envelope. From this objective function, we can derive the following iterative equations to obtain $\hat{\boldsymbol{a}}$:

$$\hat{a_i} = \frac{\int m_i(f) df}{\int \frac{\mathcal{N}(f; 400 \times i, 200)}{S(f)} df} \tag{15}$$

$$m_i(f) = \frac{a'_i \mathcal{N}(f; 400 \times i, 200)}{\sum_{\forall i} a'_i \mathcal{N}(f; 400 \times i, 200)} \tag{16}$$

where a'_i is the value estimated in the previous iteration. In this way, the bottom envelope of the spectrum $S(f)$ is obtained as $g(f, \hat{\boldsymbol{a}})$.

Among the various unvoiced consonants, unvoiced fricative sounds tend to have frequency components concentrated in a particular frequency band of the spectrum. We therefore detect the fricative sounds by using the ratio of the power of that band to the power of most other bands. Since the sampling rate in our current implementation is 16 kHz, we deal with only the unvoiced fricative phoneme /SH/ because we found from our observations that the other unvoiced fricative phonemes tended to have much power in the frequency region above 8 kHz, which is the Nyquist frequency of 16-kHz sampling. Since the phoneme /SH/ has strong power in the frequency region from 6 to 8 kHz, we define the existence degree of the phoneme /SH/ as follows:

$$E_{\mathrm{SH}} = \frac{\int_{6000}^{8000} g(f, \hat{\boldsymbol{a}}) df}{\int_{1000}^{8000} g(f, \hat{\boldsymbol{a}}) df}. \tag{17}$$

Regions in which E_{SH} is below a threshold (0.4) are identified as *nonexistence* regions, where phoneme /SH/ does not exist. The threshold 0.4 was determined experimentally. Note that to avoid any effect from bass drums, we do not use frequency components below 1 kHz in the calculation of E_{SH}.

D. Viterbi Alignment

In this section, we describe our method of executing Viterbi alignment between lyrics and separated signals. We first create a language model from the given lyrics and then extract feature vectors from separated vocal signals. Finally, we execute the Viterbi alignment between them. We also describe our method of adapting a phone model to the specific singer of the input audio signals.

1) Lyrics Processing Using the Filler Model: Given the lyrics corresponding to input audio signals, we create a phoneme network for forced alignment. This network basically does not have a branch. By using this network as a language model of a speech recognition system and calculating the most likely path of a sequence of the feature vectors extracted from the audio signals based on the Viterbi search algorighm, the start and end times of each node of network, which correspond to a phoneme in the lyrics, can be estimated. Note that nodes in the network are replaced by the HMMs of corresponding phonemes in the phoneme model. Thus, we can align the lyrics with the audio signals. In our system, since we only have the phoneme model for the Japanese language, English phonemes are substituted with the most similar Japanese phoneme.

We first convert the lyrics to a sequence of phonemes and then create a phoneme network by using the following rules:

- convert the boundary of a sentence or phrase into multiple appearances of short pauses (SPs);
- convert the boundary of a word into one appearance of an SP.

Fig. 6 shows an example of conversion from lyrics to the language model.

Some singers often sing words and phrases not in the actual lyrics, such as “Yeah” and “La La La,” during interlude sections and rests between phrases in the lyrics. We found in our preliminary experiments that such inter-phrase vowel utterances reduced the accuracy of the system because the system inevitably aligned other parts of the lyrics to those utterances. This shortcoming can be eliminated by introducing the filler model [18], [19], which is used in keyword-spotting research.

Fig. 7 is a filler model that we used in this paper. The five nodes in the figure (a, i, u, e, and o) are Japanese vowel

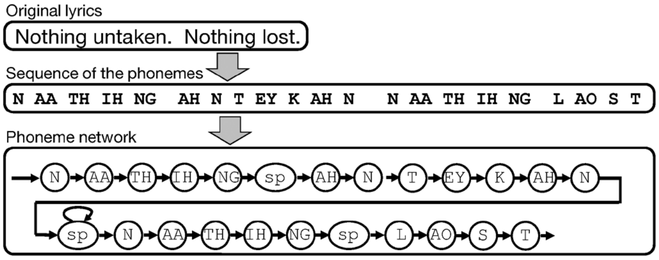

Original lyrics
Nothing untaken. Nothing lost.
Sequence of the phonemes
N AA TH IH NG AH N T EY K AH N N AA TH IH NG L AO S T
Phoneme network
N AA TH IH NG sp AH N T EY K AH N
sp N AA TH IH NG sp L AO S T

Fig. 6. Example of conversion from original lyrics to a phoneme network. Original lyrics are converted to a sequence of the phonemes first, then a phoneme network is constructed from the sequence. Note that sp represents a short pause. This lyrics was taken from the song No. 100 in RWC-MDB-P-2001.

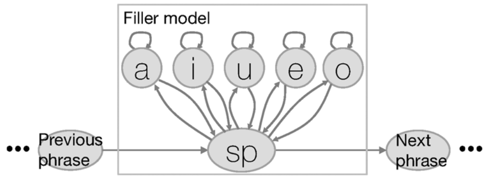


Fig. 7. Filler model inserted at each phrase boundary in the lyrics.

phonemes. This model is inserted in the middle of two consecutive phrases in the phoneme network. For example, in Fig. 6, the multiple appearance of sp between the 12th phoneme and 13th phoneme (both are /NN/) will be replaced by the filler model in Fig. 7. If there are utterances that are not written in the lyrics at that part, vowel nodes of the filler model (a, i, u, e, and o) appear here and reduce the influence of such utterances. On the other hand, if there is not such utterance vowel nodes of the filler model (a, i, u, e, and o) are ignored and the most likely path connects the two phrases via the /SP/ model.

In our preliminary experiments without using this filler model, we expected the SPs to represent short nonvocal sections. However, if the singer sang words not in the lyrics in nonvocal sections, the SPs, which were originally trained using nonvocal sections, were not able to represent them. Thus, lyrics from other parts were incorrectly allocated to these nonvocal sections. The vowels from the filler model can cover these inter-phrase utterances.

2) Adaptation of a Phone Model: We adapt a phone model to the specific singer of input audio signals. As an initial phone model, we use a monophone model for speech, since creating a phone model for a singing voice from scratch requires a large annotated training database and this type of a database of singing voices has not yet been developed. Our adaptation method consists of the following three steps:

Step 1) adapt a phone model for clean speech to a clean singing voice;

Step 2) adapt the phone model for a clean singing voice to the singing voice separated using the accompaniment sound reduction method;

Step 3) adapt the phone model for separated speech to the specific singer of input audio signals by using the unsupervised adaptation method.

Steps 1 and 2 are carried out preliminarily, and step 3 is carried out at runtime.

As an adaptation method, we use MLLR [20] and MAP [21], which are commonly used in speech recognition research. We manually annotated phoneme labels to the adaptation data for supervised adaptation. Fig. 8 shows an example of phoneme labeling.

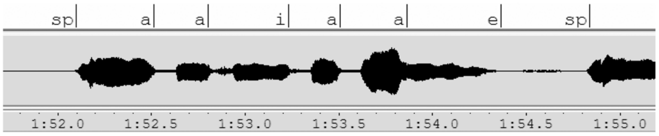


Fig. 8. Example of phoneme labeling.

TABLE I
EVALUATION DATA SET

Song #	Singer	Gender
12	Kazuo Nishi	Male
27	Shingo Katsuta	Male
32	Masaki Kuehara	Male
37	Hatae Yoshinori	Male
39	Kousuke Morimoto	Male
7	Tomomi Ogata	Female
13	Konbu	Female
20	Eri Ichikawa	Female
65	Makiko Hattori	Female
75	Hiromi Yoshii	Female

3) Alignment: Using the language model created from the given lyrics, the feature vectors extracted from separated vocal signals, and the adapted phone model for the specific singers, we execute the Viterbi alignment (forced alignment). In this alignment process, we do not allow any phoneme except /SP/ to appear in the nonvocal region and do not allow the phoneme /SH/ to appear in the region of fricative sound nonexistence. MFCCs [22] and derivatives of the MFCCs and power are used as feature vectors for the Viterbi alignment.

III. EXPERIMENTS

A. Experimental Condition

The performance of our system was evaluated experimentally. As an evaluation data set ten Japanese songs by ten singers (five male, five female) were used as listed in Table I. The songs were taken from the "RWC Music Database: Popular Music" (RWC-MDB-P-2001) [23]. They were largely in Japanese, but some phrases in their lyrics were in English. In these experiments, English phonemes were approximated by using similar Japanese phonemes. We conducted a five-fold cross-validation.

We used as the training data for the vocal activity detection method 19 songs also taken from the RWC-MDB-P-2001, sung by the 11 singers listed in Table II. These singers differed from the singers used for evaluation. We applied the accompaniment sound reduction method to the training data and we set η_{fixed} to 1.5.

Table III shows the analysis conditions for the Viterbi alignment. As an initial phone model, we used the gender-independent monophone model developed by the IPA Japanese Dictation Free Software Project and Continuous Speech Recognition Consortium (CSRC) [24]. To convert the lyrics to a sequence of phonemes, we used Mecab [25], which is a Japanese morphological analysis system.

The evaluation was performed by using phrase level alignment. In these experiments, we defined a phrase as a section that was delimited in the original lyrics by a space or a line feed.

TABLE II
TRAINING DATA FOR VOCAL ACTIVITY DETECTION

Singer	Gender	Song #
Hiroshi Sekiya	M	48, 49, 51
Katsuyuki Ozawa	M	15, 41
Masashi Hashimoto	M	56, 57
Satoshi Kumasaka	M	47
Oriken	M	6
Tomoko Nitta	F	26
Kaburagi Akiko	F	55
Yuzu Iijima	F	60
Reiko Sato	F	63
Tamako Matsuzaka	F	70
Donna Burke	F	81, 89, 91, 93, 97

TABLE III
CONDITIONS FOR ANALYSIS OF VITERBI ALIGNMENT

Sampling	16 kHz, 16 bit
Window function	Hamming
Frame length	25 ms
Frame period	10 ms
Feature vector	12th-order MFCC 12th-order ΔMFCC ΔPower

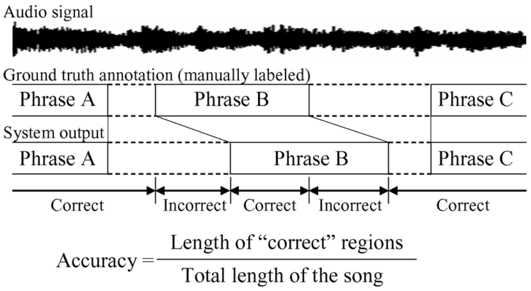


Fig. 9. Evaluation measure in the experiments on the synchronization of music and lyrics.

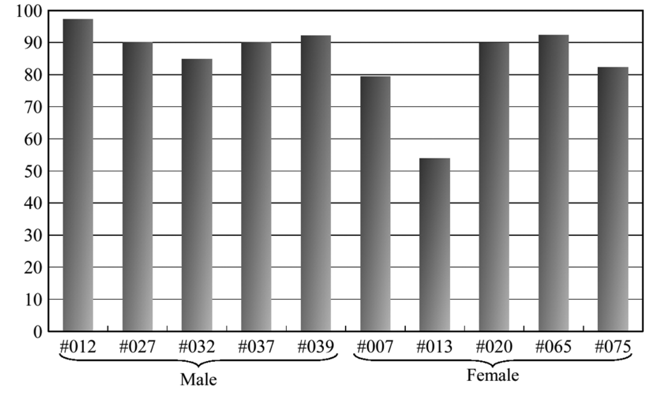


Fig. 10. Experimental results: evaluation of the whole system.

The evaluation measure we used was the ratio of the total length of the sections labeled correctly at the phrase level to the total length of a song (Fig. 9).

B. Evaluation of the Whole System

We conducted experiments using a system in which all of the methods described in this paper were implemented. Fig. 10 shows the results of these experiments.

When we compare the results in Fig. 10 between male and female singers, we see that the accuracy for the females is lower. This is because it is hard to capture the characteristics of voices with a high F0 [26]. Analyzing the errors in each song, we found that errors typically occurred at the sections in which the lyrics were sung in English. Using similar Japanese phonemes to approximate English phonemes thus seemed to be difficult. To overcome this problem, we will try to use an English phone model in combination with a Japanese one.

TABLE IV
EXPERIMENTAL RESULTS (%): EVALUATION OF ACCOMPANIMENT SOUND REDUCTION METHOD

Song #*	Gender	1) With reduction method	2) Without reduction method
12	Male	97.6	89.9
27	Male	87.2	65.1
32	Male	58.9	58.8
37	Male	93.1	93.9
39	Male	87.1	87.2
7	Female	50.4	75.7
13	Female	50.7	48.4
20	Female	60.3	52.4
65	Female	75.3	35.4
75	Female	60.8	66.2
Average		72.1	67.3

*Song number in the RWC-MDB-P-2001.

In addition to the above evaluation, we also conducted another evaluation based on a morpheme label ground truth to see how well the system performed at the morpheme level. We prepared morpheme label annotations for songs No. 12 and No. 20, calculated the accuracies using the results of the above experiment. An evaluation measure is the same as that explained in Fig. 9, except that morphemes were used instead of phrases. The accuracy for No. 12 was 72.4% and that for No. 20 was 65.3%. From these results, we can see that our system still achieved performance above 65%, though there was a certain number of inevitable decreases.

C. Evaluation of Accompaniment Sound Reduction Method

In our experimental evaluation of the accompaniment sound reduction, we disabled the vocal activity detection, fricative detection, and filler model and we enabled the three-step adaptation. We compared the following two conditions: 1) MFCC extracted from segregated singing voice using accompaniment sound reduction method and 2) MFCC extracted directly from polyphonic music without using accompaniment sound reduction method. Note that condition 1) in this experiment is the same as condition 4) in the experiment in Section III-F. We can see in Table IV that the accompaniment sound reduction improved the accuracy by 4.8 percentage points.

D. Evaluation of Vocal Activity Detection, Fricative Detection, and Filler Model

The purpose of this experiment was to investigate the separate effectiveness of the fricative detection, filler model, and vocal activity detection. We tested our method under five conditions.

1) **Baseline**: Only the three-step adaptation was enabled.
2) **VAD**: Only vocal activity detection and the three-step adaptation were enabled (Section II-B3).
3) **Fricative detection**: Only fricative sound detection and the three-step adaptation were enabled (Section II-C).

TABLE V
EXPERIMENTAL RESULTS (%): EVALUATION OF FRICATIVE DETECTION, FILLER MODEL AND VOCAL ACTIVITY DETECTION

Song #*	Gender	1) Baseline	2) VAD	3) Fric.	4) Filler	5) Proposed
12	Male	97.6	97.8	97.6	97.2	97.2
27	Male	87.2	90.2	87.2	87.9	90.1
32	Male	58.9	81.3	64.7	60.9	84.8
37	Male	93.1	91.5	93.0	92.4	90.2
39	Male	87.1	93.9	87.1	88.8	92.1
7	Female	50.4	79.9	51.6	51.0	79.4
13	Female	50.7	50.3	50.7	52.8	53.9
20	Female	60.3	92.7	60.3	62.7	90.0
65	Female	75.3	91.2	75.3	76.1	92.2
75	Female	60.8	82.0	60.8	61.1	82.3
Average		72.1	85.1	72.8	73.1	85.2

*Song number in RWC-MDB-P-2001.

4) **Filler model**: Only filler model and the three-step adaptation were enabled (Section II-D1).
5) **Proposed**: The fricative-sound detection, the filler model, the vocal-activity detection, and the three-step adaptation were enabled.

We see in Table V that vocal-activity detection, the fricative detection, and the filler model increased the average accuracy by 13.0, 0.7, and 1.0 percentage points, respectively, and that the highest accuracy, 85.2%, was obtained when all three were used. The vocal activity detection was the most effective of the three techniques. Inspection of the system outputs obtained with the filler model showed that the filler model was effective not only for utterances not in the actual lyrics, but also for non-vocal regions that could not be removed by vocal activity detection. Since our evaluation measure was phrase-based, the effectiveness of fricative detection could not be fully evaluated. Inspection of the phoneme-level alignment results showed that phoneme gaps in the middle of phrases were shorter than they were without fricative detection. We plan to develop a measure for evaluating phoneme-level alignment.

E. Evaluation of Feature Vector for Vocal Activity Detection

In our experimental evaluation of the feature vectors for vocal activity detection, we disabled the fricative detection and filler model and enabled the three-step adaptation. We compared the effectiveness of 1) the novel feature vector based on the power of harmonic structure described in Section II-B3 with that of the LPMCC-based feature vector proposed in [6]. We also compare receiver operating characteristic (ROC) curves of these two conditions. Note that condition 2) in this experiment is same as the condition 3) in the experiment in Section III-D.

We can see in Table VI that the accuracy obtained with the novel feature vector proposed in this paper was 4.0 percentage points better than that obtained with the LPMCC-based feature vector.

Fig. 11 shows the ROC curves of our vocal activity detection system. By changing the application-dependent threshold, η_{fixed}, to the various values, various pairs of the hit rates and false alarm rates are plotted. The vertical and horizontal axes represents the hit rate and false alarm rate, respectively. Note that these rates are calculated by using all the ten songs. Also from this figure, we can see that our new feature vector improves the accuracy of vocal activity detection.

TABLE VI
EXPERIMENTAL RESULTS (%): EVALUATION OF ACCOMPANIMENT SOUND REDUCTION METHOD

Song #*	Gender	1) New feature vector	2) LPMCC and ΔF0
12	Male	97.8	95.7
27	Male	90.2	87.4
32	Male	81.3	66.4
37	Male	91.5	83.7
39	Male	93.9	93.6
7	Female	79.9	62.8
13	Female	50.3	63.6
20	Female	92.7	93.3
65	Female	91.2	73.7
75	Female	82.0	90.6
Average		85.1	81.1

*Song number in RWC-MDB-P-2001.

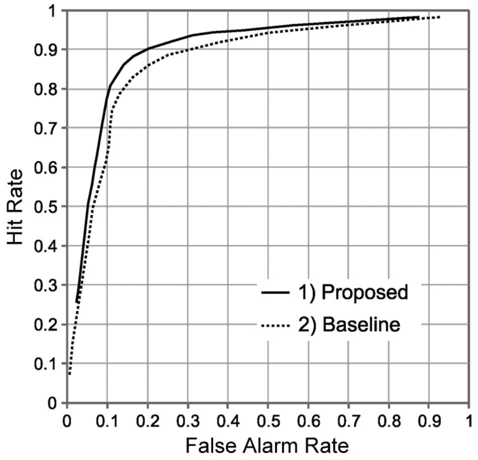


Fig. 11. Comparison of ROC curves.

F. Evaluation of Adaptation Method

In our experimental evaluation of the effectiveness of the adaptation method, we disabled the vocal activity detection, fricative detection, and filler model and we conducted experiments under the following four conditions.

1) **No adaptation**: We did not execute phone model adaptation.
2) **One-step adaptation**: We adapted a phone model for clean speech directly to separated vocal signals. We did not execute an unsupervised adaptation to input audio signals.
3) **Two-step adaptation**: First, we adapted a phone model for clean speech to clean vocal signals, and then we adapted the phone model to separated vocal signals. We did not execute an unsupervised adaptation to input audio signals.
4) **Three-step adaptation (proposed)**: First, we adapted a phone model for clean speech to clean vocal signals, then we adapted the phone model to separated vocal signals, and finally we adapted the phone model to the specific singer of input audio signals.

We can see in Table VII that our adaptation method was effective for all ten songs.

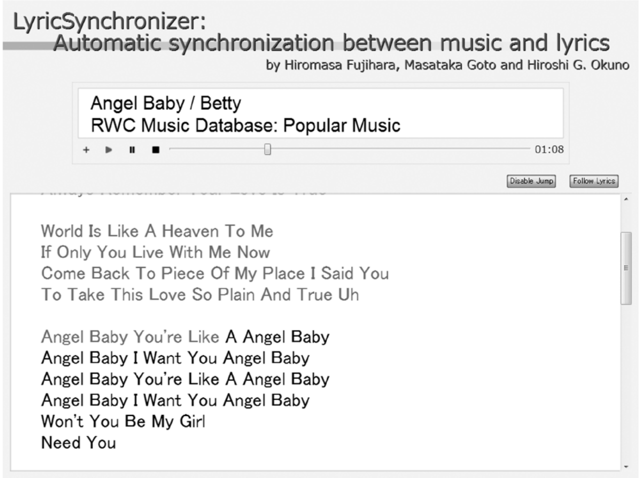
IV. LYRICSYNCHRONIZER: MUSIC PLAYBACK INTERFACE WITH SYNCHRONIZED-LYRICS-DISPLAY

Using our method for synchronizing music and lyrics, we developed a music playback interface called *LyricSynchronizer*. This interface can display the lyrics of the song synchronized

TABLE VII
EXPERIMENTAL RESULTS (%): EVALUATION OF ADAPTATION METHOD

Song #*	Gender	1) No adaptation	2) 1 step	3) 2 steps	4) 3 steps
12	Male	51.8	95.8	89.6	97.6
27	Male	23.5	70.5	82.5	87.2
32	Male	32.7	54.7	51.0	58.9
37	Male	63.8	94.0	95.1	93.1
39	Male	59.8	84.7	92.0	87.1
7	Female	13.3	42.9	49.6	50.4
13	Female	1.6	31.7	52.4	50.7
20	Female	35.0	34.9	65.7	60.3
65	Female	29.1	72.0	72.2	75.3
75	Female	17.6	44.6	49.1	60.8
Average		32.8	62.6	69.9	72.1

*A song number of the RWC-MDB-P-2001.


LyricSynchronizer:
Automatic synchronization between music and lyrics
by Hiromasa Fujihara, Masataka Goto and Hiroshi G. Okuno

Angel Baby / Betty
RWC Music Database: Popular Music

World Is Like A Heaven To Me
If Only You Live With Me Now
Come Back To Piece Of My Place I Said You
To Take This Love So Plain And True Uh

Angel Baby You're Like A Angel Baby
Angel Baby I Want You Angel Baby
Angel Baby You're Like A Angel Baby
Angel Baby I Want You Angel Baby
Won't You Be My Girl
Need You

Fig. 12. Screenshot of our music playback interface.

with the music playback. It also has a function that enables users to jump to a phrase of interest by clicking on the lyrics. Fig. 12 shows a screen shot of the interface.

The diffusion of the personal computer and the portable music player has increased our opportunities to listen to songs while using devices that have a display. It might be natural to consider using that display to enrich users' experience in music appreciation. Most devices with a display show bibliographic information such as the name of the song and the performing artist, and music players on personal computers sometimes have visualizer functions that display animations created from the spectrum of the music.

Focusing on lyrics as information that should be displayed, we developed a music playback interface that has the following two functions: displaying-synchronized-lyrics function, and jump-by-clicking-the-lyrics function. The former function displays the current position of the lyrics as shown in Fig. 12. Although this function resembles the lyrics display for karaoke, manually labeled temporal information is required in the lyrics display in karaoke. By the latter function, users can change the current playback position by clicking a phrase in the lyrics that are displayed. This function is useful when users want to listen only to sections of interest to them. This function can be considered as an implementation of *active music listening interfaces* [27].

V. CONCLUSION

We have described a system for automatically synchronizing musical audio signals and their corresponding lyrics. For accurate synchronization we segregate the singing voice and the accompaniment sound. We also developed a robust phoneme network using a filler model and developed methods for detecting vocal activity and fricative sound detection for adapting a phoneme model to the separated vocal signals of a specific singer. Experimental results showed that our system can accurately synchronize musical audio signals and their lyrics.

In our vocal activity detection method, the tradeoff between hit rate and correct rejection rate can be adjusted by changing a parameter. Although the balance between hit rate and correct rejection rate differs depending on the application, little attention has been given to this tradeoff in past research. Our vocal activity detection method makes it possible to adjust the tradeoff based on Otsu's method [13]. The novel feature vectors based on the F0 and the power of harmonic components were robust to high-pitched sounds because a spectral envelope did not need to be estimated. The underlying idea of the fricative detection (i.e., the detection of *nonexistence* regions) is a novel one. Experimental evaluation showed that synchronization performance was improved by integrating this information, even if it was difficult to accurately detect each fricative sound. Although the filler model is a simple idea, it worked very efficiently because it did not allow a phoneme in the lyrics to be skipped and it appeared only when it was needed. We proposed a method for adapting a phone model for speech to separated vocal signals. This method was useful for music and lyric alignment as well as for recognizing lyrics in polyphonic music.

We plan to incorporate higher-level information such as song structures and thereby achieve more advanced synchronization between music and lyrics. We also plan to expand our music playback interface, LyricSynchronizer, by incorporating other element of music besides lyrics and develop more advanced active music listening interfaces that can enhance music listening experiences of users.

REFERENCES

[1] Y. Wang, M.-Y. Kan, T. L. Nwe, A. Shenoy, and J. Yin, "Lyrically: Automatic synchronization of acoustic musical signals and textual lyrics," in *Proc. 12th ACM Int. Conf. Multimedia*, 2004, pp. 212–219.

[2] C. H. Wong, W. M. Szeto, and K. H. Wong, "Automatic lyrics alignment for Cantonese popular music," *Multimedia Syst.*, vol. 4–5, no. 12, pp. 307–323, 2007.

[3] A. Loscos, P. Cano, and J. Bonada, "Low-delay singing voice alignment to text," in *Proc. Int. Comput. Music Conf. (ICMC99)*, 1999.

[4] C.-K. Wang, R.-Y. Lyu, and Y.-C. Chiang, "An automatic singing transcription system with multilingual singing lyric recognizer and robust melody tracker," in *Proc. 8th Euro. Conf. Speech Commun. Technol. (Eurospeech'03)*, 2003, pp. 1197–1200.

[5] M. Gruhne, K. Schmidt, and C. Dittmar, "Phoneme recognition in popular music," in *Proc. 8th Int. Conf. Music Inf. Retrieval (ISMIR'07)*, 2007, pp. 369–370.

[6] H. Fujihara, M. Goto, T. Kitahara, and H. G. Okuno, "A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity based music information retrieval," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 638–648, Mar. 2010.

[7] M. Goto, "A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals," *Speech Commun.*, vol. 43, no. 4, pp. 311–329, 2004.

[8] M. Bay, A. F. Ehmann, and J. S. Downie, “Evaluation of multiple-F0 estimation and tracking systems,” in *Proc. 10th Int. Soc. Music Inf. Retrieval Conf. (ISMIR’09)*, 2009, pp. 315–320.

[9] G. E. Poliner, D. P. Ellis, A. F. Ehmann, E. Gómez, S. Streich, and B. Ong, “Melody transcription from music audio: Approaches and evaluation,” *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 4, pp. 1247–1256, May 2007.

[10] A. L. Berenzweig and D. P. W. Ellis, “Locating singing voice segments within music signals,” in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust. (WASPAA)*, 2001, pp. 119–122.

[11] W.-H. Tsai and H.-M. Wang, “Automatic detection and tracking of target singer in multi-singer music recordings,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’04)*, 2004, pp. 221–224.

[12] T. L. Nwe and Y. Wang, “Automatic detection of vocal segments in popular songs,” in *Proc. 5th Int. Conf. Music Inf. Retrieval (ISMIR’04)*, 2004, pp. 138–145.

[13] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Trans. System, Man, Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[14] X. R. T. Galas, “Generalized functional approximation for source filter system modeling,” in *Proc. 2nd Eur. Conf. Speech Commun. Technol. (Eurospeech’91)*, 1991, pp. 1085–1088.

[15] K. Tokuda, T. Kobayashi, and S. Imai, “Adaptive cepstral analysis of speech,” *IEEE Trans. Speech Audio Process.*, vol. 3, no. 6, pp. 481–489, Nov. 1995.

[16] T. Kitahara, M. Goto, and H. G. Okuno, “Pitch-dependent identification of musical instrument sounds,” *Appl. Intell.*, vol. 23, no. 3, pp. 267–275, 2005.

[17] H. Kameoka, M. Goto, and S. Sagayama, “Selective amplifier of periodic and non-periodic components in concurrent audio signals with spectral control envelopes,” 2006, vol. 2006, pp. 77–84, IPSJ SIG Tech. Rep., no. 90.

[18] R. E. Méliani, “Accurate keyword spotting using strictly lexical fillers,” in *Proc. 1997 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’97)*, 1997, pp. 907–910.

[19] A. S. Manos and V. W. Zue, “A segment-based wordspotter using phonetic filler models,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’97)*, 1997, pp. 899–902.

[20] J. L. Gauvain and C.-H. Lee, “Maximum a posteriori estimation for multivariate Gaussian mixture observations of Markov chains,” *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 291–298, Apr. 1994.

[21] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Comput. Speech Lang.*, vol. 9, pp. 171–185, 1995.

[22] S. B. Davis and P. Mermelstein, “Comparison of parametric representation for monosyllabic word recognition,” *IEEE Trans. Acoustic, Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[23] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, “RWC music database: Popular, classical, and jazz music databases,” in *Proc. 3rd Int. Conf. Music Inf. Retrieval (ISMIR’02)*, Oct. 2002, pp. 287–288.

[24] T. Kawahara, A. Lee, K. Takeda, and K. Shikano, “Recent progress of open-source LVCSR engine Julius and Japanese model repository—Software of continuous speech recognition consortium—,” in *Proc. 6th Int. Conf. Spoken Lang. Process. (Interspeech’04 ICSLP)*, 2004.

[25] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying conditional random fields to Japanese morphological analysis,” in *Proc. Conf. Empirical Methods in Natural Lang. Process.*, 2004, pp. 230–237.

[26] A. Sasou, M. Goto, S. Hayamizu, and K. Tanaka, “An auto-regressive, non-stationary excited signal parameter estimation method and an evaluation of a singing-voice recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’05)*, 2005, pp. I-237–I-240.

[27] M. Goto, “Active music listening interfaces based on signal processing,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP’07)*, 2007, pp. IV-1441–IV-1444.

Hiromasa Fujihara received the Ph.D. degree from Kyoto University, Kyoto, Japan, in 2010 for his work on computational understanding of singing voices.

He is currently a Research Scientist with the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His research interests include singing information processing and music information retrieval.

Dr. Fujihara was awarded the Yamashita Memorial Research Award from the Information Processing Society of Japan (IPSJ).

Masataka Goto received the Doctor of Engineering degree from Waseda University, Tokyo, Japan, in 1998.

He is currently the leader of the Media Interaction Group, National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. He serves concurrently as a Visiting Professor at the Institute of Statistical Mathematics, an Associate Professor (Cooperative Graduate School Program) in the Graduate School of Systems and Information Engineering, University of Tsukuba, and a Project Manager of the MITOH Program (the Exploratory IT Human Resources Project) Youth division by the Information Technology Promotion Agency (IPA).

Dr. Goto received 25 awards over the past 19 years, including the Commendation for Science and Technology by the Minister of MEXT ‘‘Young Scientists’ Prize,’’ the DoCoMo Mobile Science Awards ‘‘Excellence Award in Fundamental Science,’’ the IPSJ Nagao Special Researcher Award, and the IPSJ Best Paper Award.

Jun Ogata received the B.E., M.E., and Ph.D. degrees in electronic and information engineering from Ryukoku University, Kyoto, Japan, in 1998, 2000, and 2003, respectively.

He is currently a Research Scientist with the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. His research interests include automatic speech recognition, spoken language understanding, and speech-based interface.

Hiroshi G. Okuno (M’03–SM’06) received the B.A. and Ph.D. degrees from the University of Tokyo, Tokyo, Japan, in 1972 and 1996, respectively.

He worked for NTT, JST, and Tokyo University of Science. He is currently a Professor of the Graduate School of Informatics, Kyoto University, Kyoto, Japan. He was a Visiting Scholar at Stanford University, Stanford, CA, from 1986 to 1988. He has done research in programming languages, parallel processing, and reasoning mechanism in AI. He is currently engaged in computational auditory scene analysis, music scene analysis, and robot audition. He co-edited *Computational Auditory Scene Analysis* (Lawrence Erlbaum Associates, 1998), *Advanced Lisp Technology* (Taylor & Francis, 2002), and *New Trends in Applied Artificial Intelligence (IEA/AIE)* (Springer, 2007).

Prof. Okuno received various awards including the 1990 Best Paper Award of JSAI, the Best Paper Award of IEA/AIE-2001, 2005, and 2010, IEEE/RSJ IROS-2010 NTF Award for Entertainment Robots and Systems, and IROS-2001 and 2006 Best Paper Nomination Finalist. He is a member of AAAI, ACM, ASJ, ISCA, and 5 Japanese societies.