AN ADAPTIVE KARAOKE SYSTEM THAT PLAYS ACCOMPANIMENT PARTS OF MUSIC AUDIO SIGNALS SYNCHRONOUSLY WITH USERS' SINGING VOICES

Yusuke Wada **Yoshiaki Bando** **Eita Nakamura** **Katsutoshi Itoyama** **Kazuyoshi Yoshii**
Department of Intelligence Science and Technology
Graduate School of Informatics, Kyoto University, Japan
{wada, yoshiaki, enakamura, itoyama, yoshii}@sap.ist.i.kyoto-u.ac.jp

ABSTRACT

This paper presents an adaptive karaoke system that can extract accompaniment sounds from music audio signals in an online manner and play those sounds synchronously with users' singing voices. This system enables a user to expressively sing an arbitrary song by dynamically changing the tempo of the user's singing voices. A key advantage of this systems is that users can immediately enjoy karaoke without preparing musical scores (MIDI files). To achieve this, we use online methods of singing voice separation and audio-to-audio alignment that can be executed in parallel. More specifically, music audio signals are separated into singing voices and accompaniment sounds from the beginning using an online extension of robust nonnegative matrix factorization. The separated singing voices are then aligned with a user's singing voices using online dynamic time warping. The separated accompaniment sounds are played back according to the estimated warping path. The quantitative and subjective experimental results showed that although there is room for improving the computational efficiency and alignment accuracy, the system has a great potential for offering a new singing experience

1. INTRODUCTION

Karaoke is one of the most popular ways of enjoying music in which people can sing their favorite songs synchronously with musical accompaniment sounds prepared in advance. In the current karaoke industry, musical scores (MIDI files) are assumed to be available for generating accompaniment sounds. Professional music transcribers are therefore asked to manually transcribe music every time new commercial CD recordings are released. The critical issues of this approach are that music transcription is very time-consuming and technically demanding and that the quality of accompaniment sounds generated from MIDI files is inferior to that of original musical audio signals.

It is impractical for the conventional approach to manually transcribe a huge number of songs on the Web. Consumer generated media (CGM) has recently been become more and more popular and many non-professional people have composed and distributed their own original songs on the Web. In Japan, for example, over 120 thousand songs have been uploaded on a media sharing Web service from July 2007 [1]. It is thus necessary to generate high-quality accompaniment sounds from *arbitrary* music audio signals without using musical scores or lyrics.

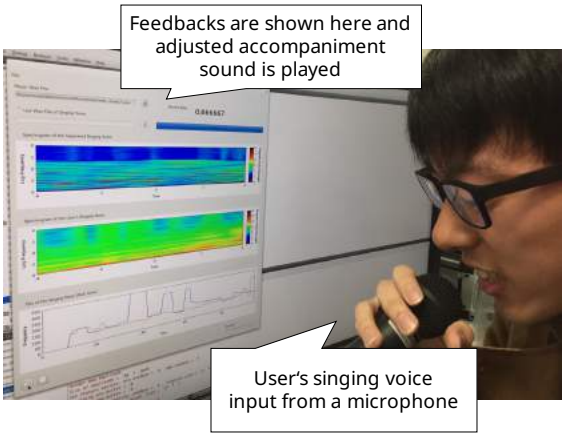

Figure 1. An example of how to use the proposed system. A user is allowed to expressively sing a song while accompaniment sounds are played back synchronously with the user's singing voices. The spectrograms and F0 trajectories of the user's singing voices and those of the time-stretched original singing voices can be compared in real time. The progress of singing voice separation is also displayed.

Another limitation of the current karaoke systems is that users need to manually set the tempo of accompaniment sounds in advance. Although this limitation can be acceptable for standard popular music with steady tempo, some kinds of music (e.g., opera, gospel, and folk songs) are usually sung in an expressive way by dynamically changing the tempo of the music.

To solve these problems, we propose an adaptive karaoke system that can extract accompaniment sounds from music audio signals in an online manner and play those sounds synchronously with users' singing voices.[1] Figure 1 shows how to use the proposed system. Once a song is selected, a user is allowed to immediately start to sing the song while listening to adaptively played-back accompaniment sounds separated from music audio signals. If the user gradually

[1] A demo video of the proposed system is available online: http://sap.ist.i.kyoto-u.ac.jp/members/wada/smc2017/index.html

accelerates (decelerates) the singing, the tempo of accompaniment sounds is accelerated (decelerated) accordingly such that those sounds are synchronized with the user's singing. The pitches (fundamental frequencies, F0s) of the singing voices can be compared with those of original singing voices in real time. To use this system, all the user has to do is to prepare only music audio signals.

This system mainly consists of three components: karaoke generation based on singing voice separation, audio-to-audio singing-voice alignment, and time-stretching of accompaniment sounds (Figure 2). More specifically, accompaniment sounds are separated from music audio signals using an online extension of robust nonnegative matrix factorization (RNMF) [2]. The stretch rate of the separated accompaniment sounds is estimated using online dynamic time warping (DTW) between a user's and original singing voices. Finally, the stretched version of the accompaniment sounds is played back. Since these steps are in parallel, it can therefore conceal the processing time for the singing voice separation from the user.

The main technical contribution of this study is to tackle real-time audio-to-audio alignment between singing voices whose pitches, timbres, and tempos many significantly vary over time. Note that conventional studies on singing-voice alignment focus on alignment between singing voices and symbolic information such as musical scores or lyrics. Another contribution is to apply this fundamental technique to a practical application of music performance assistance.

2. RELATED WORK

This section reviews related work on singing information processing and automatic accompaniment.

2.1 Karaoke Systems

Tachibana *et al.* [3] proposed a karaoke system that generates accompaniment sounds from input music audio signals without preparing musical scores or lyrics. This system uses a voice suppression technique to generate the accompaniment sounds, whose pitches can be changed manually. Inoue *et al.* [4] proposed another karaoke system that automatically adjusts the tempo of accompaniment sounds to a user's singing voices, assuming that musical scores and lyrics are prepared in advance.

2.2 Automatic Music Accompaniment

There have been many studies on automatic music accompaniment [5–11]. Dannenberg [5] proposed an online algorithm based on dynamic programming for automatic accompaniment. Vercoe [6] proposed an accompaniment system that supports live performances using traditional musical instruments. Raphael [7] used a hidden Markov model (HMM) to find optimal segmentation of the musical score of a target musical piece. Cont [8] designed an architecture that features two coupled audio and tempo agents based on a hybrid hidden Markov/semi-Markov framework. Nakamura *et al.* [9] reduced the computational complexity of polyphonic MIDI score following using an outer-product HMM. Nakamura *et al.* [10] also proposed an efficient score-following algorithm under an assumption that the prior distributions of score positions before and after repeats or skips are independent of each other. Montecchio and Cont [11] proposed a particle-filter-based method of real-time audio-to-audio alignment between polyphonic audio signals without using musical scores.

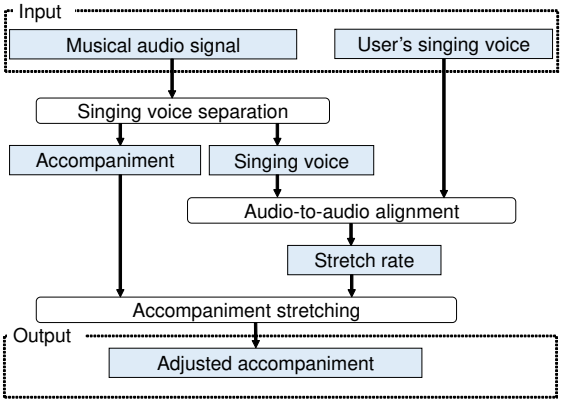

Figure 2. An overview of the system implementation.

2.3 Singing Voice Alignment

Many studies have addressed audio-to-score or audio-to-lyric alignment, where singing voices are aligned with symbolic data such as musical scores or lyrics [12–15]. Gong *et al.* [12] attempted audio-to-score alignment based on a hidden semi-Markov model (HSMM) using melody and lyric information. A lot of effort has been devoted to audio-to-lyric alignment. Fujihara *et al.* [13], for example, used singing voice separation and phoneme alignment for synchronizing musical audio signals with their corresponding lyrics. Iskandar *et al.* [14] attempted syllabic-level alignment based on dynamic programming. Wang *et al.* [15] combined feature extraction from singing voices rhythmic structure analysis of musical audio signals. Dzhambazov *et al.* [16] modeled a duration of each phoneme based on a duration-explicit HMM using mel-frequency cepstral coefficients (MFCCs).

2.4 Singing Voice Separation

A typical approach to singing voice separation is to estimate a time-frequency mask that separates the spectrogram of a target music audio signal into a vocal spectrogram and an accompaniment spectrogram [17–20]. Huang *et al.* [17] used robust principal component analysis (RPCA) to extract accompaniment spectrograms with low-rank structures. Deep recurrent neural networks were also used [21]. Ikemiya *et al.* [18] improved the separation quality by combining RPCA with F0 estimation. Rafii and Pardo [19] proposed a similarity-based method to find repetitive patterns (accompaniment sounds) in polyphonic audio signals. As another approach, Yang *et al.* [20] used Bayesian non-negative matrix factorization (NMF). Very few studies have been conducted on online singing voice separation.

3. PROPOSED SYSTEM

This section describes the graphical user interface (GUI) of the proposed system and the implementation of the system based on singing voice separation and audio-to-audio alignment between singing voices.

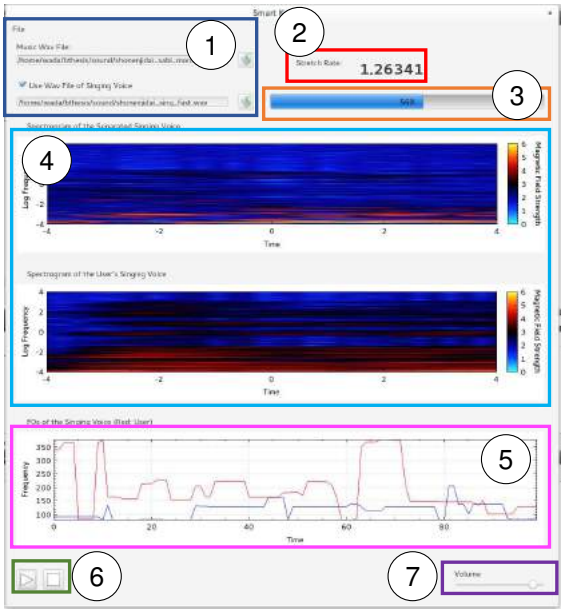

Figure 3. A screenshot of the user interface.

3.1 User Interface

Figure 3 shows a screenshot of the user interface that provides easy-to-use functions through seven GUI elements: (1) a selector for music audio signals, (2) a display of the current stretch rate, (3) a display of the progress of singing voice separation, (4) a display of the spectrograms of the user's and the original singing voices, (5) a display of the F0 trajectories of the user's and the original singing voices, (6) play and stop buttons for controlling the playback, and (7) a volume control for the accompaniment sound.

The GUI elements numbered 2, 4, and 5 provides visual feedback of the user's singing voice and the original singing voice. The red area (number 2 in Figure 3) indicates whether the stretch rate is matching the user's intention. The user can refer how the original singer sings with the spectrograms displayed in the sky blue area (number 4 in Figure 3). For example, the user can know sections in which the original singer users a vibrato technique. In addition, the F0 trajectories displayed in the pink area (number 5 in Figure 3) helps the user to correct the pitch of the user's singing voice.

3.2 Implementation Policies

To reduce the user's wait time, we specify three requirements for the system implementation. First, users should be able to enjoy karaoke immediately after starting the system. Second, singing voice separation should to be processed in real time without prior learning. Third, automatic accompaniment should also be processed in real time.

We chose and implemented a method for each component of the system so as to satisfy these three requirements. More specifically, singing voice separation, recording of a user's singing voices, singing-voice alignment, and playback of time-stretched accompaniment sounds are processed in independent threads (Figure 2).

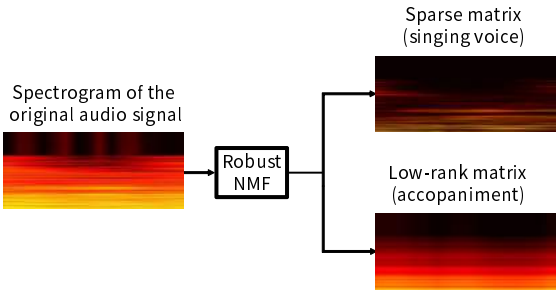

Figure 4. Singing voice separation based on VB-RNMF. The matrix corresponding to an input audio spectrogram is separated into a sparse matrix corresponding to the magnitude spectrogram of singing voices and a low-rank matrix corresponding to the magnitude spectrogram of accompaniment sounds.

3.3 Singing Voice Separation for Music Audio Signals

To separate a musical audio signal specified by the user into singing voices and accompaniment sounds, we propose an online version of variational Bayesian robust NMF (VB-RNMF) [2]. Although there are many offline methods of singing voice separation [17–20], our system requires real-time separation in order to conceal the processing time of the singing voice separation from the user. Figure 4 shows how online VB-RNMF separates a mini-batch spectrogram into a sparse singing-voice spectrogram and a low-rank accompaniment spectrogram.

More specifically, an input spectrogram $\mathbf{Y} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_T]$ is approximated as the sum of a low-rank spectrogram $\mathbf{L} = [\boldsymbol{l}_1, \ldots, \boldsymbol{l}_T]$ and a sparse spectrogram $\mathbf{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T]$:

$$\boldsymbol{y}_t \approx \boldsymbol{l}_t + \boldsymbol{s}_t, \tag{1}$$

where $\mathbf{L}$ is represented by the product of K spectral basis vectors $\mathbf{W} = [\boldsymbol{w}_1, \ldots, \boldsymbol{w}_K]$ and their temporal activation vectors $\mathbf{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_T]$ as follows:

$$\boldsymbol{y}_t \approx \mathbf{W}\boldsymbol{h}_t + \boldsymbol{s}_t. \tag{2}$$

The tread-off between low-rankness and sparseness is controlled in a Bayesian manner stated below.

The Kullback-Leibler (KL) divergence is used for measuring the approximation error. Since the maximization of the Poisson likelihood (denoted by $\mathcal{P}$) corresponds to the minimization of the KL divergence, the likelihood function is given by

$$p(\mathbf{Y}|\mathbf{W}, \mathbf{H}, \mathbf{S}) = \prod_{f,t} \mathcal{P}\left(y_{ft} \middle| \sum_k w_{fk}h_{kt} + s_{ft}\right). \tag{3}$$

Since the gamma distribution (denoted by $\mathcal{G}$) is a conjugate prior for the Poisson distribution, gamma priors are put on the basis and activation matrices of the low-rank components as follows:

$$p(\mathbf{W}|\alpha^{wh}, \beta^{wh}) = \prod_{f,k} \mathcal{G}(w_{fk}|\alpha^{wh}, \beta^{wh}), \tag{4}$$

$$p(\mathbf{H}|\alpha^{wh}, \beta^{wh}) = \prod_{k,t} \mathcal{G}(h_{kt}|\alpha^{wh}, \beta^{wh}), \tag{5}$$

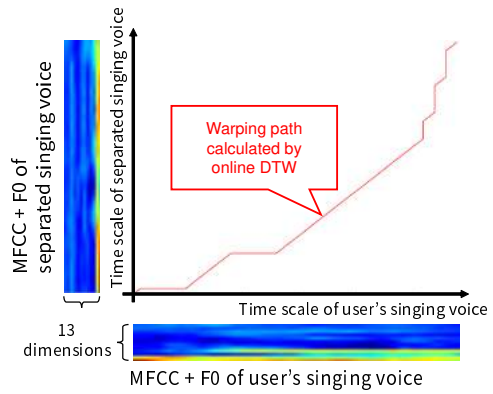

Figure 5. A warping path obtained by online DTW.

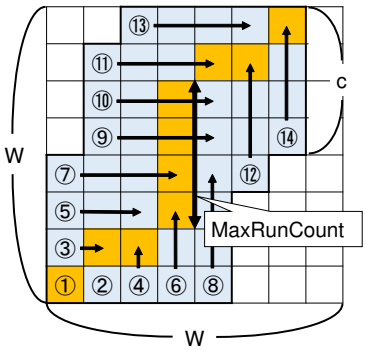

Figure 6. Online DTW with input length $W = 8$, search width $c = 4$, and path constraint $MaxRunCount = 4$. All calculated cells are framed in bold and colored sky blue, and the optimal path is colored orange.

where α^{wh} and β^{wh} are represent the shape and rate parameters of the gamma distribution.

To force the sparse components to take nonnegative values, gamma priors with rate parameters given the Jeffreys hyperpriors are put on those components as the following:

$$p(\mathbf{S}|\alpha^s, \boldsymbol{\beta}^s) = \prod_{f,t} \mathcal{G}(s_{ft}|\alpha^s, \beta^s_{ft}), \tag{6}$$

$$p(\beta^s_{ft}) \propto (\beta^s_{ft})^{-1}. \tag{7}$$

where α^s represents the hyperparameter of the gamma distribution that controls the sparseness. Using Eqs. (3)–(7), the expected values of $\mathbf{W}$, $\mathbf{H}$, and $\mathbf{S}$ are estimated in a mini-batch style using a VB technique.

3.4 Audio-to-Audio Alignment between Singing Voices

We use an online version of dynamic time warping (DTW) [22] that estimates an optimal warping path between a user's singing voices and original singing voices separated from music audio signals (Figure 5). Since the timbres and F0s of the user's singing voices can significantly differ from those of the original singing voices according to the singing skills of the user, we focus on both MFCCs and F0s of singing voices for calculating a cost matrix in DTW. To estimate the F0s, saliency-based F0 estimation called subharmonic summation [23] is performed.

First MFCCs and F0s are extracted from the mini-batch spectrogram of the separated voice $\boldsymbol{X} = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_W\}$ and that of the user's singing voice $\boldsymbol{Y} = \{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_W\}$. Suppose we represent the concatenated vector of MFCCs and F0 extracted from the $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ as $\boldsymbol{x}'_i = \{\boldsymbol{m}^{(x)}_i, f^{(x)}_i\}$ and $\boldsymbol{y}'_i = \{\boldsymbol{m}^{(y)}_i, f^{(y)}_i\}$, $\boldsymbol{X}' = \{\boldsymbol{x}'_1, \cdots \boldsymbol{x}'_W\}$ and $\boldsymbol{Y}' = \{\boldsymbol{y}'_1, \cdots \boldsymbol{y}'_W\}$ are input to the online DTW. In this concatenation, the weight of F0 is smaller than MFCCs. This is because F0 would be much less stable than MFCCs when the user has poor skills. If those mini-bathes are not silent, the MFCCs and F0s are extracted and the cost matrix $D = \{d_{i,j}\}(i = 1, \cdots, W; j = 1, \cdots, W)$ is updated according to Algorithms 1 and 2 with constraint parameters W, c, and $MaxRunCount$, *i.e.*, a partial row or column of D is calculated as follows:

$$d_{i,j} = ||\boldsymbol{x}'_i - \boldsymbol{y}'_j|| + \min \begin{cases} d_{i,j-1} \\ d_{i-1,j} \\ d_{i-1,j-1}. \end{cases} \tag{8}$$

The variable s and t in Algorithms 1 and 2 represent the current position in the feature sequences $\boldsymbol{X}'$ and $\boldsymbol{Y}'$ respectively. The online DTW calculates the optimal warping path $L = \{\boldsymbol{o}_1, \cdots, \boldsymbol{o}_l\}, \boldsymbol{o}_i = (i_k, j_k)$ $(0 \leq i_k \leq i_{k+1} \leq n; 0 \leq j_k \leq j_{k+1} \leq n)$, using the root mean square for $||\boldsymbol{x}_i - \boldsymbol{y}_j||$ incrementally, without backtraking. (i_k, j_k) means that the frame $\boldsymbol{x}_{i_k}$ corresponds to the frame $\boldsymbol{y}_{j_k}$. Figure 6 shows an example how the cost matrix and the warping path is calculated. Each number in Figure 6 represents the order to calculate the cost matrix. The parameter W is the length of the input mini-batch. If the warping path reaches the W-th row or column, then the calculation stops. If the warping path ends at $(W, k)(k < W)$, the next warping path starts from that point. c restricts the calculation of the cost matrix. At most c successive elements are calculated for each calculation of the cost matrix. $MaxRunCount$ restricts the shape of the warping path. The warping path is incremented at most $MaxRunCount$.

The function $GetInc$ decides which to increment a row, column, or both of the warping path. If the row is incremented from the position (s, t) in the cost matrix, then at most c successive elements from $(s - c, t)$ to (s, t) are calculated. Otherwise, successive elements from $(s, t - c)$ to (s, t) are calculated.

For the system reported here, we set the parameters as $W = 300, c = 4, MaxRunCount = 3$.

3.5 Time Stretching of Accompaniment Sounds

Given the warping path $L = \{\boldsymbol{o}_1, \cdots, \boldsymbol{o}_l\}$, the system calculates a series of stretch rates $R = \{r_1, \cdots, r_W\}$ for each frame of a mini-batch. The stretch rate of the i-th frame, r_i, is given by

$$r_i = \frac{\text{the amount of } i \text{ in } \{i_1, \cdots, i_l\}}{\text{the amount of } i \text{ in } \{j_1, \cdots, j_l\}}. \tag{9}$$

Algorithm 1 The online DTW algorithm

$s \leftarrow 1, t \leftarrow 1$, path $\leftarrow (s,t)$, previous $\leftarrow$ None
Calculate $d_{s,t}$ following the eq. 8
while $s < W, t < W$ **do**
  if GetInc$(s,t) \neq$ Column **then**
    $s \leftarrow s+1$
    for $k = t-c+1, \cdots, t$ **do**
      if $k > 0$ **then**
        Calculate $d_{s,t}$ following the eq. 8
      end if
    end for
  end if
  if GetInc$(s,t) \neq$ Row **then**
    $t \leftarrow t+1$
    for $k = s-c+1, \cdots, s$ **do**
      if $k > 0$ **then**
        Calculate $d_{s,t}$ following the eq. 8
      end if
    end for
  end if
  if GetInc$(s,t) ==$ previous **then**
    runCount $\leftarrow$ runCount $+1$
  else
    runCount $\leftarrow 1$
  end if
  if GetInc$(s,t) \neq$ Both **then**
    previous $\leftarrow$ GetInc(s,t)
  end if
  path.append$((s,t))$
end while

Algorithm 2 The function $GetInc(s,t)$

if $s < c$ **then**
  return Both
end if
if runCount $<$ MaxRunCount **then**
  if previous $==$ Row **then**
    return Column
  else
    return Row
  end if
end if
$(x,y) = \arg\min(D(k,l))$, where $k == s$ or $l == t$
if $x < s$ **then**
  return Row
else if $y < t$ **then**
  return Column
else
  return Both
end if

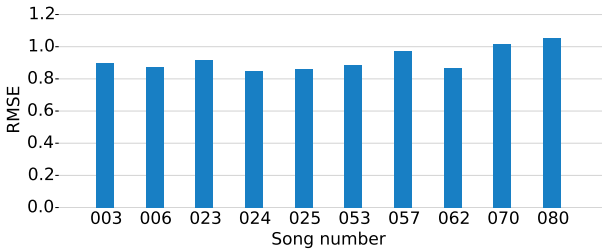

Figure 7. Stretch rate RMSEs measured in the accuracy evaluation. The RMSEs represent how the estimated stretch rates differ from the original rates.

Then the stretch rate r for the current mini-batch is calculated as the mean of R as $r = \frac{1}{W}\sum_{i=1}^{W} r_i$. r is updated on each iteration of the online DTW. The system finally uses a standard method of time-scale modification called phase vocoder [24] to stretch the mini-batch of the separated accompaniment sounds by a factor of r. The phase vocoder stretches the input sound globally by a factor of r.

4. EVALUATION

We conducted three experiments to evaluate the effectiveness of this proposed system. Quantitatively, we evaluated the efficiency of the system performance and the accuracy of real-time audio-to-audio alignment. We also conducted a subjective experiment.

4.1 Efficiency Evaluation of Singing Voice Separation

To evaluate the efficiency of singing voice separation, we used 100 pieces sampled at 44.1 kHz from the RWC popular music database [25]. Each piece was truncated to 30 seconds from the beginning. Spectrograms were then calculated using short-term Fourier transform (STFT) with the window size of 4096 samples and the hop size of 10 ms. Each spectrogram was then split into 30 300-millisecond mini-batches, which were input to online VB-RNMF.

The average processing time for a 300-millisecond mini-batch was 538.93 ms, and no mini-batches were processed in less than 300 ms. This means that the singing voice separation actually does not work in real time, but it is sufficient to wait a short while before using the system. For greater convenience, the performance of singing voice separation could be improved. One way to achieve this is to process singing voice separation on a graphics processing unit (GPU).

4.2 Accuracy Evaluation of Singing Voice Alignment

To evaluate the accuracy of audio-to-audio alignment, we randomly selected 10 pieces from the database. The singing voices were separated from the 30-second spectrogram of each piece. The phase vocoder [24] was then used to stretch the separated singing voices according to eleven kinds of stretch rates, $r = 0.5, 0.6, \cdots, 1.4, 1.5$. The separated voice and the stretched version of it were input to online DTW, and the stretch rate r' was calculated from the estimated warping path. Then the r' and r were compared. The system uses the separated singing voice and the user's clean voice, but this evaluation uses the separated singing voice and a stretched version of the voice to determine a correct stretch rate.

Figure 7 shows the stretch rate root mean squared errors (RMSEs) between r and r'. The average RMSE over 10 pieces was 0.92 and the standard deviation was 0.068. This indicates that the performance difference of audio-to-audio alignment varied little over different songs, but the align-

	question (1)	question (2)
subject 1	partially yes	partially yes
subject 2	yes	partially yes
subject 3	partially yes	no
subject 4	yes	partially yes

Table 1. The result of the subjective evaluation.

ment accuracy was not very high. This was because the separated singing voices contained musical noise, and the time stretch led to further noise, giving inaccurate MFCCs.

The number of songs used in this evaluation is rather small. We plan further evaluation using more songs.

There are many possibilities for improving the accuracy. First, HMM-based methods would be superior to the DTW-based method. HMM-based methods learn previous inputs unlike DTW-based methods, and this would be useful to improve the accuracy. Second, simultaneous estimation of the alignment and the tempo estimation would improve the accuracy. The result of tempo estimation could help to predict the alignment. An approach for this way of alignment is a method using particle filters [11].

4.3 Subjective Evaluation

Each of four subjects was asked to sing a Japanese popular song after listening to the songs in advance. The songs used for evaluation were an advertising jingle "Hitachi no Ki," a rock song "Rewrite" by Asian Kung-fu Generation, a popular song "Shonen Jidai" by Inoue Yosui, and a popular song "Kimagure Romantic" by Ikimono Gakari. The subjects were then asked two questions; (1) whether the automatic accompaniment was accurate, and (2) whether the user interface was appropriate.

The responses by the subjects are shown in Table 1. The responses indicate, respectively, that the automatic accompaniment was partially accurate and practical, and the user interface was useful.

The subjects also gave several opinions for the system. First, the accompaniment sounds were low quality and it was not obvious whether the automatic accompaniment was accurate. We first need to evaluate quality of the singing voice separation. An approach for this problem could be to add a mode of playing a click sound according to the current tempo. Second, some of the subjects did not understand what the displayed spectrograms represented. Some explanation should be added for further user-friendliness, or only the stretch rate and F0 trajectories should be displayed.

The number of the test sample used in this subjective evaluation is rather small. We plan further evaluation by more subjects.

5. CONCLUSION

This paper presented a novel adaptive karaoke system that plays back accompaniment sounds separated from music audio signals while adjusting the tempo of those sounds to that of the user's singing voices. The main components of the system are singing voice separation based on online VB-RNMF and audio-to-audio alignment between singing voices based on online DTW. This system enables a user to expressively sing an arbitrary song by dynamically changing the tempo of the user's singing voices. The quantitative and subjective experimental results showed the effectiveness of the system.

We plan to improve separation and alignment of singing voices. Using the tempo estimation result would help improvement of the audio-to-audio alignment. Automatic harmonization for users' singing voices would be an interesting function as a smart karaoke system. Another important research direction is to help users improve their singing skills by analyzing the weak points from the history of the matching results between the user's and original singing voices.

Acknowledgments

This study was supported by JST OngaCREST and OngaACCEL Projects and JSPS KAKENHI Nos. 26700020, 24220006, 26280089, 16H01744, and 15K16654.

6. REFERENCES

[1] M. Hamasaki *et al.*, "Songrium: Browsing and listening environment for music content creation community," in *Proc. SMC*, 2015, pp. 23–30.

[2] Y. Bando *et al.*, "Variational bayesian multi-channel robust nmf for human-voice enhancement with a deformable and partially-occluded microphone array," in *Proc. EUSIPCO*, 2016, pp. 1018–1022.

[3] H. Tachibana *et al.*, "A real-time audio-to-audio karaoke generation system for monaural recordings based on singing voice suppression and key conversion techniques," *J. IPSJ*, vol. 24, no. 3, pp. 470–482, 2016.

[4] W. Inoue *et al.*, "Adaptive karaoke system: Human singing accompaniment based on speech recognition," in *Proc. ICMC*, 1994, pp. 70–77.

[5] R. B. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proc. ICMC*, 1984, pp. 193–198.

[6] B. Vercoe, "The synthetic performer in the context of live performance," in *Proc. ICMC*, 1984, pp. 199–200.

[7] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden markov models," *J. IEEE Trans. on PAMI*, vol. 21, no. 4, pp. 360–370, 1999.

[8] A. Cont, "A coupled duration-focused architecture for realtime music to score alignment," *J. IEEE Trans. on PAMI*, vol. 32, no. 6, pp. 974–987, 2010.

[9] E. Nakamura *et al.*, "Outer-product hidden markov model and polyphonic midi score following," *J. New Music Res.*, vol. 43, no. 2, pp. 183–201, 2014.

[10] T. Nakamura *et al.*, "Real-time audio-to-score alignment of music performances containing errors and arbitrary repeats and skips," *J. IEEE/ACM TASLP*, vol. 24, no. 2, pp. 329–339, 2016.

[11] N. Montecchio *et al.*, "A unified approach to real time audio-to-score and audio-to-audio alignment using sequential montecarlo inference techniques," in *Proc. ICASSP*, 2011.

[12] R. Gong *et al.*, "Real-time audio-to-score alignment of singing voice based on melody and lyric information," in *Proc. Interspeech*, 2015.

[13] H. Fujihara *et al.*, "LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics," in *Proc. IEEE Journal of Selected Topics in Signal Processing Conference*, 2011, pp. 1252–1261.

[14] D. Iskandar *et al.*, "Syllabic level automatic synchronization of music signals and text lyrics," in *Proc. ACMMM*, 2006, pp. 659–662.

[15] Y. Wang *et al.*, "LyricAlly: Automatic synchronization of textual lyrics to acoustic music signals," *J. IEEE TASLP*, vol. 16, no. 2, pp. 338–349, 2008.

[16] G. Dzhambazov *et al.*, "Modeling of phoneme durations for alignment between polyphonic audio and lyrics," in *Proc. SMC*, 2015, pp. 281–286.

[17] P.-S. Huang *et al.*, "Singing-voice separation from monaural recordings using robust principal component analysis," in *Proc. IEEE ICASSP*, 2012, pp. 57–60.

[18] Y. Ikemiya *et al.*, "Singing voice separation and vocal F0 estimation based on mutual combination of robust principal component analysis and subharmonic summation," *J. IEEE/ACM TASLP*, vol. 24, no. 11, pp. 2084–2095, 2016.

[19] Z. Rafii *et al.*, "Music/voice separation using the similarity matrix," in *Proc. ISMIR*, 2012, pp. 583–588.

[20] P.-K. Yang *et al.*, "Bayesian singing-voice separation," in *Proc. ISMIR*, 2014, pp. 507–512.

[21] P.-S. Huang *et al.*, "Singing-voice separation from monaural recordings using deep recurrent neural networks," in *Proc. ISMIR*, 2014, pp. 477–482.

[22] S. Dixon, "An on-line time warping algorithm for tracking musical performances," in *Proc. the 19th IJCAI*, 2005, pp. 1727–1728.

[23] D. J. Hermes, "Measurement of pitch by subharmonic summation," *J. ASA*, vol. 83, no. 1, pp. 257–264, 1988.

[24] J. Flanagan *et al.*, "Phase vocoder," *Bell System Technical Journal*, vol. 45, pp. 1493–1509, 1966.

[25] M. Goto *et al.*, "Rwc music database: Popular, classical, and jazz music databases," in *Proc. ISMIR*, 2002, pp. 287–288.