SPEAKBYSINGING: CONVERTING SINGING VOICES TO SPEAKING VOICES WHILE RETAINING VOICE TIMBRE

Shimpei Aso†, *Takeshi Saitou*‡, *Masataka Goto*‡, *Katsutoshi Itoyama*†,
Toru Takahashi†, *Kazunori Komatani*†, *Tetsuya Ogata*†, *Hiroshi G. Okuno*†

† Graduate School of Informatics, Kyoto University
‡ National Institute of Advanced Industrial Science and Technology (AIST)
aso [at] kuis.kyoto-u.ac.jp

ABSTRACT

This paper describes a singing-to-speaking synthesis system called *"SpeakBySinging"* that can synthesize a speaking voice from an input singing voice and the song lyrics. The system controls three acoustic features that determine the difference between speaking and singing voices: the fundamental frequency (F0), phoneme duration, and power (volume). By changing these features of a singing voice, the system synthesizes a speaking voice while retaining the timbre of the singing voice. The system first analyzes the singing voice to extract the F0 contour, the duration of each phoneme of the lyrics, and the power. These features are then converted to target values that are obtained by feeding the lyrics into a traditional text-to-speech (TTS) system. The system finally generates a speaking voice that preserves the timbre of the singing voice but has speech-like features. Experimental results show that SpeakBySinging can convert singing voices into speaking voices whose timbre is almost the same as the original singing voices.

1. INTRODUCTION

The goal of this research is to synthesize attractive speaking voices by controlling the acoustic features unique to them. Most previous research approaches, such as concatenative synthesis [1, 2] and Hidden Markov Model (HMM)-based synthesis [3, 4, 5], have focused on *text-to-speech synthesis*, which generates a speaking voice from scratch when given the text. In contrast, our approach focuses on *singing-to-speech synthesis*, which converts a voice singing any text (e.g., the lyrics of a song) into a speaking voice. Research on the singing-to-speech synthesis is important for investigating the acoustic differences between singing and speaking voices. It will also be useful for manipulating singing voices while retaining their timbre. In addition, singing-to-speech synthesis itself is interesting for end users because even if we do not have actual recordings of a singer's speaking voice, we can arrange for the singer to speak "virtually" by using this synthesis technique.

Among the previous approaches, Saitou *et al.* [6] proposed a speech-to-singing synthesis system called *"SingBySpeaking"* that converts a speaking voice to a singing voice. Based on acoustic differences between singing and speaking voices, SingBySpeaking synthesizes a singing voice by providing acoustic features that are unique to singing voices to an input speaking voice. The synthesized singing voice is natural and reflects the timbre of the speaking voice. Our "singing-to-speech synthesis" approach was inspired by this speech-to-singing synthesis, and corresponds to an inverted version of it. Because the nature of the target signals is different, however, we cannot simply use the same technique for this inverted process. In fact, this paper describes new techniques that are necessary to convert a singing voice to a speaking voice, which cannot be achieved using the approach by Saitou *et al.* [6].

We propose a novel speaking voice synthesis system, *"SpeakBySinging"*, that can convert a singing voice to a speaking voice while retaining the timbre of the voice. First, SpeakBySinging extracts three acoustic features, the F0 contour, the duration of each phoneme of the lyrics, and the power, from the input singing voice. To obtain the target values of these features, SpeakBySinging uses a traditional text-to-speech (TTS) system and supplies the text of the song lyrics to obtain a speaking voice. Note that this TTS-based speaking voice is used to obtain the natural target values; SpeakBySinging then changes the three acoustic features of the input singing voice to the target values, and finally synthesizes the speaking voice while retaining the timbre of the singing voice. Since singing voices tend to have a unique and beautiful timbre due to the different vocalism that occurs while singing compared to that while speaking, the converted speaking voices that have the timbre of singing voices will extend the variety of synthesized voices that can be obtained.

2. SINGING VOICE AND SPEAKING VOICE

In this section, we describe the acoustic differences between singing voices and speaking voices.

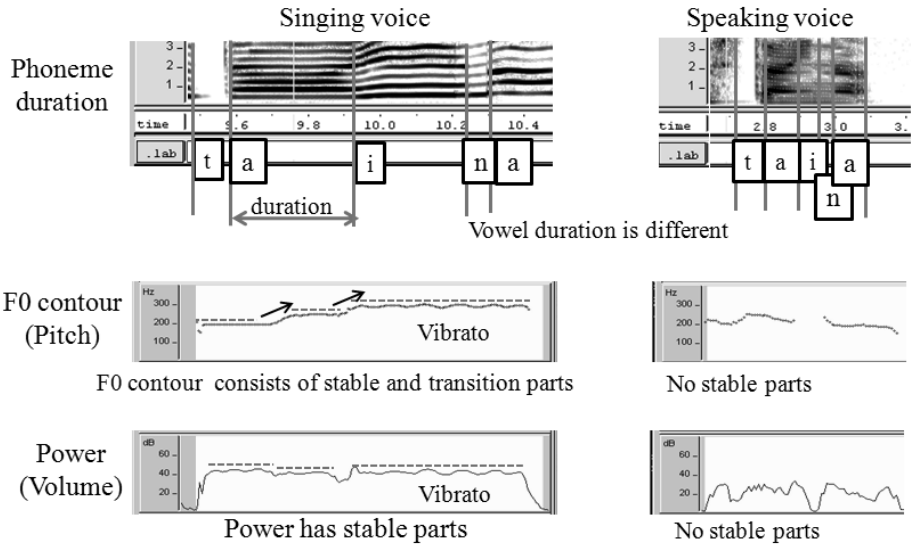

Figure 1: *Differences between singing voice and speaking voice while uttering "/t a i n a/" from the aspects of* phoneme duration*,* F0 (pitch) contour*, and* power (volume)*.*

2.1. Differences in acoustic features

Conventional studies [7, 8] focus on three points: *phoneme duration*, *F0 contour*, and *power*, to clarify the differences between singing and speaking voices. These differences are explained below using the example shown in Figure 1.

Phoneme duration For the singing voice, the duration of each phoneme changes in accordance with the musical score. For the speaking voice, on the other hand, the duration of each phoneme has a relatively similar length. To be precise, corresponding consonant parts have approximately the same length, the boundary between a consonant and a succeeding vowel also have approximately the same length, and vowel parts have different lengths.

F0 (pitch) contour

1. For the singing voice, a musical note corresponds to a steady state of the F0 contour. A musical score therefore corresponds to the F0 contour that has a step-like shape [7], as shown in Figure 1. For the speaking voice, the F0 contour has a fluid shape that has a low frequency at the beginning and end of each utterance.
2. There are quasi-periodic modulations referred to as vibrato in the F0 contour of singing voices.
3. The mean F0 of singing voices is higher than that of the speaking voices.

Power For the singing voice, power changes are synchronized with F0. For the speaking voice, the power always varies continuously.

2.2. Speech-to-singing synthesis

Saitou *et al.* [6] proposed a system that can synthesize a singing voice from a speaking voice. Its inputs are specified as follows:

- a speaking voice reading the lyrics of a song,
- the musical score of the song, and
- synchronization information where each phoneme of the speaking voice is automatically segmented and associated with a musical note in the score.

This system integrates three types of model for controlling the F0 contour, phoneme duration, and the spectrum. When converting the speaking voice to the singing voice, the F0 contour of the singing voice is generated by adding four different fluctuations into musical notes using an F0 control model. The duration of each phoneme is stretched/shrunk according to the tempo of the song. To generate the spectral envelope of the singing voice, the following two spectrum control models are used to modify the speaking voice:

- Spectral control model 1 adds the "singer's formant," which is a remarkable spectral peak around 3 kHz [9].
- Spectral control model 2 adds amplitude modulation synchronized with F0 vibrato [10].

The singing voice is then synthesized from the modified features. This system can synthesize the singing voice while retaining the timbre of the speaking voice, but cannot convert a singing voice to a speaking voice.

3. SPEAKBYSINGING

Our singing-to-speaking synthesis system *SpeakBySinging* has the following input and output:

Input Singing voice and lyrics of the song.
Output Synthesized speaking voice.

The voice conversion is achieved by changing characteristics of the three different acoustic features, i.e., phoneme duration, F0 contour, and power, into characteristics of acoustic features generated by TTS. These three features are chosen since they are the main differences between singing and speaking voices, as discussed in Section 2. The system extracts three acoustic features from the singing voice using the speech manipulation system called *STRAIGHT* [11]. To retain the voice timbre, it is important to avoid distorting the spectral envelope of the singing voice. We therefore do not control the singer's formant because it is one of the spectral characteristics unique to the singing voice.

The system consists of three modules for controlling phoneme duration, F0 contour, and power, in a STRAIGHT-based sound decomposing and synthesis function, as shown in Figure 2. Here, we denote the features of a voice synthesized by TTS as "target" features. The steps of conversion are described as follows:

1. Analyze F0 contour, spectral envelope, and aperiodicity index from the singing voice using STRAIGHT, and the duration of each phoneme using the Viterbi alignment method [12].
2. Extract target features, i.e., each phoneme duration, F0 contour, and power generated using TTS.
3. Stretch/Shrink the duration of each phoneme according to the target phoneme duration.
4. Replace the F0 contour of the singing voice with the target one while retaining the voiced/unvoiced segments and manipulating the mean F0.
5. Adjust the power of the spectral envelope to the target power.
6. Synthesize a speaking voice from the modified F0 contour, spectral envelope, and aperiodicity index.

In more detail, SpeakBySinging first decomposes the singing voice into the F0 contour, spectral sequence, and aperiodicity index sequence using STRAIGHT:

F0 contour This corresponds to vibrations of the vocal cord. A voiced sound is represented as a positive number, and an unvoiced sound as zero.

Spectral envelope This represents the vocal tract characteristics.

Aperiodicity index This represents the power of non-periodic excitation.

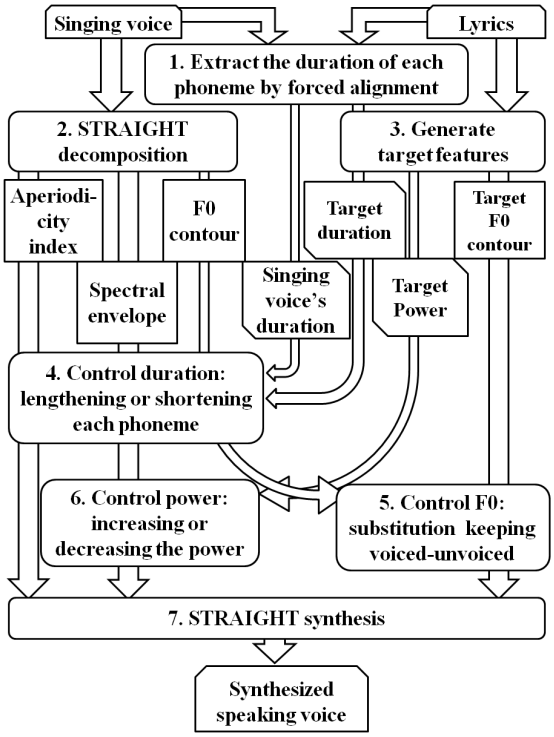

Figure 2: *Block diagram of SpeakBySinging.*

STRAIGHT provides various functions to manipulate these features independently and to synthesize the resulting waveform. These STRAIGHT functions facilitate the implementation of SpeakBySinging, since we can easily control the duration of phoneme, F0 contour, and power independently. We refer to the process of feature extraction as "STRAIGHT decomposition," and to that of synthesis as "STRAIGHT synthesis."

SpeakBySinging then estimates a temporal segment in the voice waveform that corresponds to each phoneme in the lyrics. The temporal segment (i.e., the duration) of each phoneme of a singing voice is obtained by using HMM-based Viterbi alignment [12, 13].

3.1. Target features generation

To convert a singing voice to a speaking voice, we extract features that are characteristic to a speaking voice using TTS synthesis. As explained in Section 2, these features are different between the two voices. To convert these features, we input the lyrics into a TTS system and then obtain these features of a speaking voice as target values. By controlling these singing voice features based on the target values, we can convert the features of a singing voice to those of a speaking voice.

Here, we can use any TTS system that provides: 1) du-

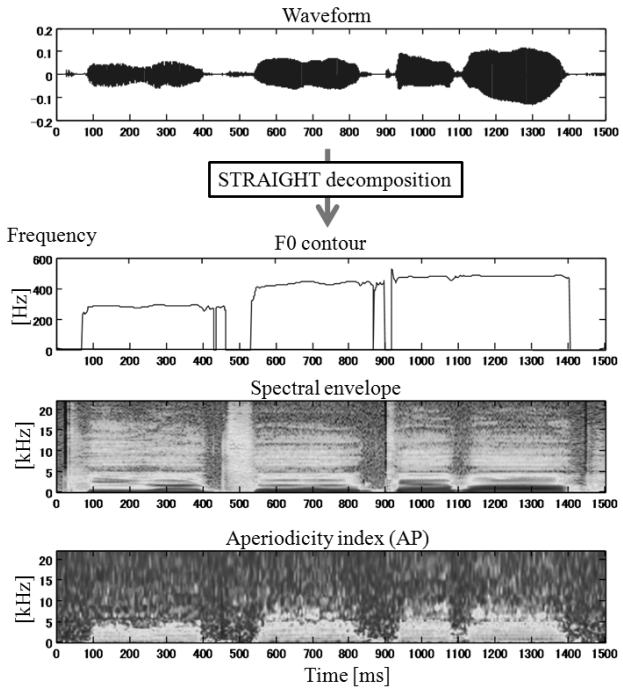

Figure 3: *STRAIGHT decomposes the input waveform and extracts an F0 contour, aperiodicity index, and spectral envelope.*

ration of each phoneme, 2) F0 contour, and 3) power. We used OpenJTalk[1], because it satisfies the above conditions.

3.2. Duration control

When a person sings, the duration of phonemes depends on the score. The duration control module stretches/shrinks each phoneme of the singing voice in order to change the duration of each phoneme to the duration for a speaking voice. The inputs and outputs of the duration control module are specified as follows:

Input Singing voice's duration, F0 contour, aperiodicity index, spectral envelope, and target duration obtained in step 2.

Output Stretched/Shrunk F0 contour, aperiodicity index, and spectral envelope.

To convert the duration of the singing voice to that of the target, we first calculate the rate of a singing voice's duration and the target duration for each phoneme. The stretch/shrink rate of the n-th phoneme, $S(n)$, is defined as:

$$S(n) = \frac{D_{(target)}(n)}{D_{(sing)}(n)} \quad (1)$$

[1] http://open-jtalk.sourceforge.net/

where $D_{(target)}(n)$ is the duration of the n-th phoneme's of the target, and $D_{(sing)}(n)$ is that of the singing voice. We then stretch/shrink STRAIGHT features to obtain a new phoneme duration, which we call the stretched/shrunk F0 contour, stretched/shrunk aperiodicity index, and stretched/shrunk spectral envelope. Linear interpolation is used to stretch/shrink the duration of the n-th phoneme using $S(n)$, as shown in Figure 4. Note that a boundary segment between a consonant and a succeeding vowel, which occupies the region ranging from 10 ms before the boundary to 30 ms after the boundary, is not stretched; only the rest is stretched/shrunk. This is because the transition time from a consonant to a vowel is approximately the same between singing and speaking voices, and thus, changing it may degrade the sound quality of the synthesized voice [6].

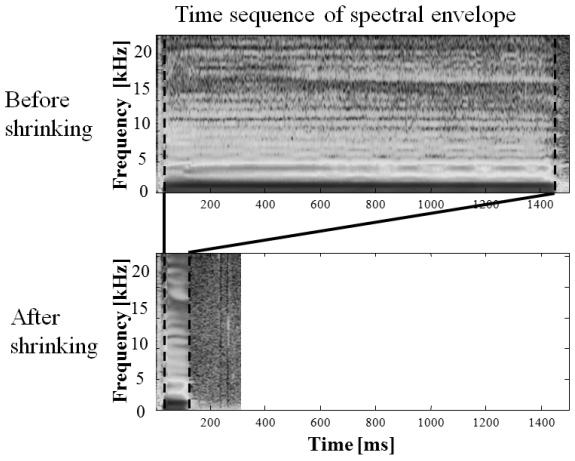

Figure 4: *Example of time stretch/shrink.*

3.3. F0 control

The F0 control module replaces the target F0 contour of the singing voice with the target F0 contour extracted from the TTS system. The inputs and output of this module are as follows:

Input Stretched/Shrunk F0 contour and the target F0 contour

Output F0 contour of the speaking voice.

The F0 control module replaces F0 contours while maintaining voiced/unvoiced segments. Here, voiced segments are those with periodic vocal tract vibrations and are expressed as segments that have positive numbers. Unvoiced segments are those without such periodicity and are expressed as segments that have zeros. These regions need to be correctly distinguished because 1) STRAIGHT alternates the synthesis process depending on whether the time

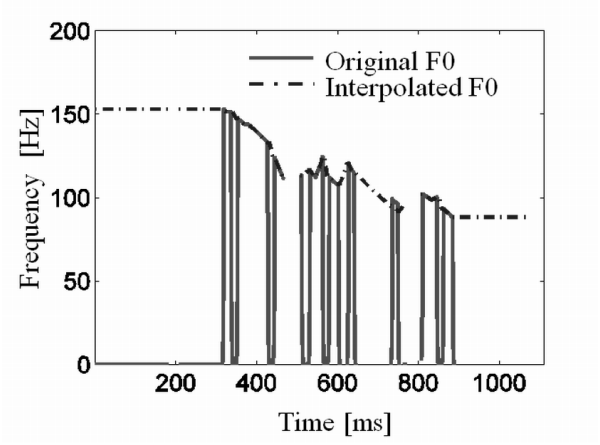

Figure 5: *Interpolation of F0 to obtain all-frames-voiced F0 contour.*

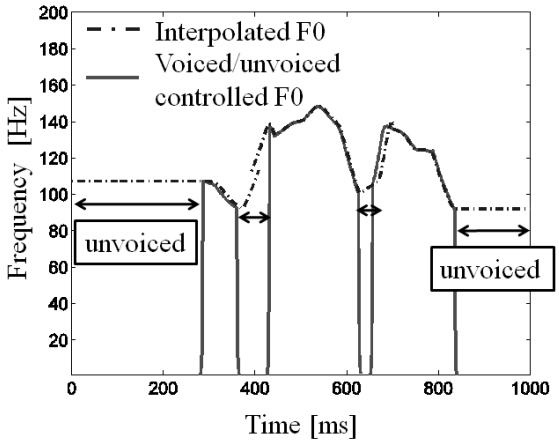

Figure 6: *Removal of F0 corresponding to unvoiced segment from all-frames-voiced F0 contour.*

frame is voiced or unvoiced, and thus, if voiced and unvoiced regions are given incorrectly, the synthesis may fail; 2) the voiced and unvoiced regions are usually different between the target F0 contour and the stretched/shrunk F0 contour. We maintain the voiced and unvoiced regions by performing the following operations:

1. We achieve this through linear interpolation and generate an F0 contour based on the target in which all time frames are assumed to be voiced, as shown in Figure 5 (we call this an all-frames-voiced F0 contour),
2. Extract unvoiced regions from the stretched/shrunk F0 contour, and
3. Obtain a new F0 contour, in which voiced segments are as those obtained in the all-frames-voiced contour, and zero everywhere else, as shown in Figure 6. We call this a VUV (voice and unvoiced region) modified target F0 contour.

The mean F0 during an utterance also needs to be set. This is because the substituted F0 contour for the speaking voice sometimes becomes much lower than that of the original singing voice, which may reduce the consistency between the F0 and the spectral envelope, resulting in unnatural voice synthesis. To avoid this inconsistency, we shift the mean F0 of the VUV modified F0 contour so that it is closer to the original F0 contour. We first calculate the mean F0 of the VUV modified F0 contour, M_{vuv}, and the mean of the stretched/shrunk F0 contour, M_{stretch}:

$$M_{\text{vuv}} = \frac{1}{\sum_{n=1}^{N} v_{\text{vuv}}(n)} \sum_{n=1}^{N} \text{F0}_{\text{vuv}}(n), \text{and} \qquad (2)$$

$$M_{\text{stretch}} = \frac{1}{\sum_{n=1}^{N} v_{\text{stretch}}(n)} \sum_{n=1}^{N} \text{F0}_{\text{stretch}}(n), \qquad (3)$$

where $\text{F0}_{\text{vuv}}(n)$ and $\text{F0}_{\text{stretch}}(n)$ are the VUV modified F0 contour and the stretched/shrunk F0 contour, respectively. Here, $v_{\text{vuv}}(n)$ and $v_{\text{stretch}}(n)$ are windows that indicate whether the time frame is voiced or not; that is:

$$v_{\text{vuv or stretch}}(n) = \begin{cases} 1, & \text{if F0}(n) > 0 \\ 0, & \text{otherwise.} \end{cases} \qquad (4)$$

Then, we shift the F0 contour by using a ratio between the above two mean F0 values and obtain a final F0 output:

$$\text{F0}_{\text{output}}(n) = \text{F0}_{\text{vuv}}(n) + \alpha v_{\text{vuv}}(n) D \qquad (5)$$

$$D = \frac{M_{\text{stretch}}}{M_{\text{vuv}}} \qquad (6)$$

Here, α is a parameter to control the mean F0. If $\alpha = 0$, there are no changes in the F0 contour. If $\alpha = 1$, the manipulated mean F0 corresponds to the mean of the stretched/shrunk F0.

3.4. Power control

The power control module modifies the power of the stretched/shrunk spectral envelope sequence, given in 3.2, and adjusts the power to that of the target. The inputs and output of this module are specified as follows:

Input Time sequence of spectral envelope and target power.

Output Time sequence of spectral envelope whose power is modified to the target.

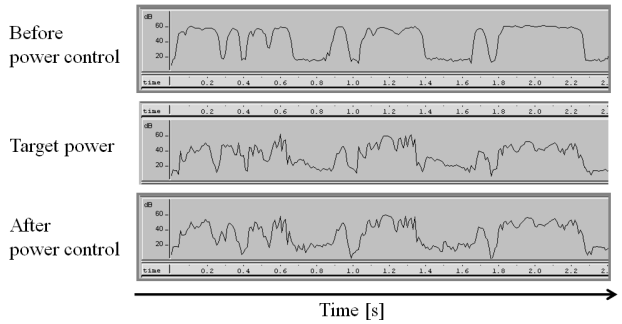

Figure 7: *Example of control input. The module extracts the power from a spectral envelope and controls it for each frame.*

An example of power control is shown in Figure 7. We define the power of the t-th time frame of the spectral envelope, $P_{\mathrm{s}}(t)$, as follows:

$$P_{\mathrm{s}}(t) = \sum_{f=1}^{F} (N_{\mathrm{s}}(f,t))^2 \tag{7}$$

Here, N_{s} is a matrix of the spectral envelope, F is the number of frequency bins, and f is the frequency index. Given the target power P_{t}, the power ratio is then defined as follows:

$$\mathrm{Ratio}(t) = 10 \log_{10} \frac{P_{\mathrm{t}}(t)}{P_{\mathrm{s}}(t)} \ [\mathrm{dB}] \tag{8}$$

We do not simply use this power ratio as obtained, but use it after applying a non-linear transfer function. This is because preliminary experiments show that when $\mathrm{Ratio}(t)$ is large, especially 15dB or larger, the consonant or the ambient noise content of the synthesized voice becomes excessive, which degrades the sound quality. Therefore, we introduce the following non-linear transfer function, which has the effect of compressing an excessively high power ratio:

$$\mathrm{Ratio\_Comp}(t) = \begin{cases} \mathrm{Ratio}(t) \\ \qquad (\mathrm{Ratio}(t) \leq \mathrm{Thre}) \\ \mathrm{Thre} + \frac{\mathrm{Ratio}(t) - \mathrm{Thre}}{\mathrm{Rate}} \\ \qquad (\mathrm{Ratio}(t) > \mathrm{Thre}) \end{cases} \tag{9}$$

where Rate is a constant that controls the rate of compression. The relationship between $\mathrm{Ratio\_Comp}$ and Ratio is plotted in Figure 8. Finally, we obtain the output time sequence of the spectral envelope, N_{o}, as follows:

$$N_{\mathrm{o}}(f,t) = N_{\mathrm{s}}(f,t) \times 10^{\frac{\mathrm{Ratio\_Comp}(t)}{20}} \tag{10}$$

4. EXPERIMENTS

We evaluated our system by conducting two psychoacoustic experiments. First, we compared the timbre of synthesized speaking and singing voices, and then evaluated the perceptual similarity in their voice timbre. Second, we evaluated the naturalness of synthesized speaking voices when the mean F0 was varied. In the experiments described as follows, we manually fixed the phoneme alignment to avoid the influence of Viterbi alignment error on the results, and we only evaluated the influence of feature manipulations.

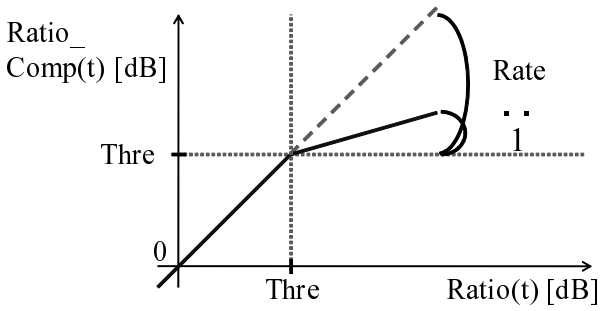

Figure 8: *Relationship between* Ratio_Comp *and* Ratio.

4.1. Evaluation of naturalness

In this experiment, we evaluated how the perceived naturalness of the synthesized sound changed as the mean F0 of the synthesized voice (parameter α mentioned in Eq. 5) was changed.

First, we obtained ten singing voice samples from the *AIST Humming Database*, which is a music database for singing research [14]. These consisted of five samples of female singers (J002, J003, J012, J014, J027) and five of male singers (J042, J048, J052, J054, J063), all of which sang the lyrics *P078_DK* (a Japanese phrase having 12 seconds). Each sample was recorded at a 16-bit resolution with a sampling frequency of 44.1 kHz, and was about 10 seconds in duration. Then, for each singing sample, six variations of speaking voice were synthesized using SpeakBySinging, each with a different value of α. Thus, 60 synthesized speech samples were prepared.

Next, we evaluated the synthesized voices with different α values using the mean opinion score (MOS). We asked 8 subjects to participate in the evaluation. All test subjects were graduate students with normal hearing ability.

Each subject listened to the synthesized voice samples, which were played at a comfortable sound pressure level using a stereo headphone (SONY MDR-CD900ST). Then, each subject was asked to assign an opinion score to each synthesized sample based on its naturalness, on a five-point scale from 1 to 5, as indicated in Table 1. We evaluated the mean of the opinion score (MOS) for samples with a particular value of α, over all test subjects. Moreover, we evaluated the MOS given for all samples synthesized from female singers, and for all samples synthesized from male

Table 1: *Description of the opinion score.*

Description	score
Highly natural	5
Natural	4
Fair	3
Unnatural	2
Highly unnatural	1

Table 2: *The MOS for different α values, for voices synthesized using female singers, male singers, and both genders.*

α	Female	Male	Both
0.0	2.10	2.65	2.38
0.2	2.43	2.60	2.43
0.4	2.31	2.33	2.31
0.6	2.29	2.40	2.29
0.8	2.23	2.30	2.22
1.0	2.18	2.20	2.18

singers. Table 2 lists the experimental results.

4.2. Evaluation of voice timbre

In this experiment, we evaluated how well the system retained the voice timbre. To do this, we synthesized speaking voices for a given set of lyrics from different singers, and asked the subjects to "match" the singing voice to the synthesized speaking voice, as depicted in Figure 9. If the timbre unique to each singer was retained, the subjects should be able to match the singing voice to the synthesized speaking voice.

For each (real) singing voice used in Section 4.1, we synthesized speaking voices of all samples using SpeakBySinging (see Figure 10). The parameter α, as defined in Eq. 5, was set to 0.6. This is a compromise between the F0 generated by the TTS ($\alpha = 0$), which is too low for female singers, and the actual singing pitch ($\alpha = 1$), which is too high for a speaking voice for both male and female singers. To isolate the effect of the STRAIGHT engine on human perception of timbral differences, we synthesized each singing voice by analyzing and directly synthesizing it using STRAIGHT.

Subjects that participated in the experiment in Section 4.1 were asked to listen to the synthesized speaking voices and the synthesized singing voices of female singers and then match the speaking voices and singing voices whose timbre sounds were the most similar, as shown in Figure 9. Next, the same subjects were asked to repeat the process, but with male singers.

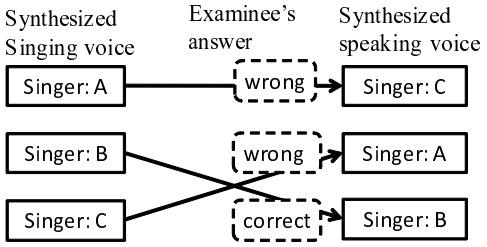

Figure 9: *A subject matched the singing voice to a speaking voice whose timbre seemed to be the same.*

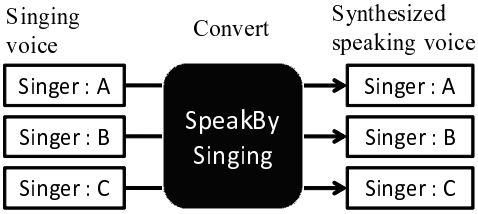

Figure 10: *Speaking voices were synthesized from singing voices.*

Finally, we computed the average *accuracy* for all test subjects, for all relevant audio pairs. We define *accuracy* as the ratio between the number of correctly matched pairs to the total number of pairs. Table 3 gives the result.

4.3. Results and discussion

In Section 4.1, the result in Table 2 shows that adjusting the mean F0 makes the synthesized voice more natural for female singers. This is because the TTS system used in SpeakBySinging specializes in male voice synthesis. Therefore, we believe that the F0 contour generated by the TTS was natural for the male singers but too low for the female singers. Hence, setting a slightly higher α affected the synthesizing of the female speaking voices.

The result in Table 3 suggests that SpeakBySinging can synthesize speaking voices while retaining the voice timbre of the singing voices. Because the average accuracy for female singers is greater than that for male, the result also suggests that the timbre of singing voices for female singers is retained better than that for male singers.

After the experiment, we conducted a survey with the participants in the experiment in order to receive other feedback. Multiple participants noted that the duration of the phonemes were unnatural. This suggests that the method for adjusting phoneme duration needs more accurate processing, such as a DTW-based alignment [15] between a singing voice and a target voice generated by TTS.

Table 3: *Accuracies of speaking voice synthesized from female singers, male, and both.*

Female	Male	Both
85.0%	72.5%	78.8%

5. CONCLUSION

We proposed SpeakBySinging, a novel system to synthesize a speaking voice from a singing voice while retaining the timbre of the singing voice. The system is based on manipulation of the F0 contour, the phoneme duration, and the power. Experimental results showed that our system is capable of retaining the timbre that is unique to a particular singer while changing aspects other than the timbre to a speaking voice.

In the future, we plan to improve the duration control and to develop a way to gradually change from a singing voice to a speaking voice in order to realize a morphing function between singing and speaking voices. Comparing the quality of the synthesized voice when the target signal is a real speaking voice instead of a TTS-synthesized speech is another future work.

6. ACKNOWLEDGMENTS

We thank Tomoyasu Nakano (AIST) for his advice on voiced/unvoiced sounds when interpolating the F0 contour and Hiromasa Fujihara (AIST) for his advice on Viterbi alignment.

This research has been partially supported by JST CrestMuse, JSPS Grants-in-Aid for Scientific Research (S), and the Kyoto University Global COE Program.

7. REFERENCES

[1] A. W. Black and N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis," in *Proc. Eurospeech*, 1995, pp. 581–584.

[2] D. Schwarz, "A system for data-driven concatenative sound synthesis," in *Proc. Digital Audio Effects*, 2000, pp. 97–102.

[3] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," *IEICE Transactions on Information and Systems (in Japanese)*, vol. J83-D2, no. 11, pp. 2099–2107, 2000.

[4] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-based singing voice synthesis system," in *Proc. International Conference on Spoken Language Processing*, 2006, pp. 1141–1144.

[5] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "A hidden semi-Markov model-based speech synthesis system," *IEICE Transactions on Information and Systems*, vol. E90-D, no. 5, pp. 825–834, 2007.

[6] T. Saitou, M. Goto, M. Unoki, and M. Akagi, "Speech-to-singing synthesis: Converting speaking voices to singing voices by controlling acoustic features unique to singing voices," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2007, pp. 215–218.

[7] Y. Ohishi, M. Goto, K. Itou, and K. Takeda, "Discrimination between singing and speaking voices," in *Proc. Eurospeech*, 2005, pp. 1141–1144.

[8] J. Sundberg, *The Science of the Singing Voice*, Northern Illinois University Press, 1987.

[9] J. Sundberg, "Articulatory interpretation of the "singing formant"," *The Journal of the Acoustical Society of America*, vol. 55, pp. 838–844, 1974.

[10] P. B. Oncley, "Frequency, amplitude, and waveform modulation in the vocal vibrato," *The Journal of the Acoustical Society of America*, vol. 49, no. 1A, pp. 136–136, 1971.

[11] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveignè, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, no. 3, pp. 187–207, 1999.

[12] H. Fujihara, M. Goto, J. Ogata, K. Komatani, T. Ogata, and H. G. Okuno, "Automatic synchronization between lyrics and music CD recordings based on Viterbi alignment of segregated vocal signals," in *Proc. IEEE International Symposium on Multimedia*, 2006, pp. 257–264.

[13] J. Odell, D. Ollason, P. Woodland, S. Young, and J. Jansen, *The HTK Book for HTK V2.0*, Cambridge University Press, Cambridge, UK, 1995.

[14] M. Goto and T. Nishimura, "AIST Humming Database: Music database for singing research," *IPSJ SIG Technical Reports (in Japanese)*, vol. 2005, no. 82, pp. 7–12, 2005.

[15] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and Information Systems*, vol. 7, no. 3, pp. 358–386, 2005.