VOCALISTENER2: A SINGING SYNTHESIS SYSTEM ABLE TO MIMIC A USER'S SINGING IN TERMS OF VOICE TIMBRE CHANGES AS WELL AS PITCH AND DYNAMICS

Tomoyasu Nakano and Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Japan

ABSTRACT

This paper presents a singing synthesis system, *VocaListener2*, that can automatically synthesize a singing voice by mimicking the timbre changes of a user's singing voice. The system is an extension of our previous *VocaListener* system which deals with only pitch and dynamics. Most previous techniques for manipulating voice timbre have focused on voice conversion and voice morphing, and they cannot deal with the timbre changes during singing. To develop VocaListener2, we constructed a *voice timbre space* on the basis of various singing voices that are synchronized under pitch, dynamics, and phoneme by using VocaListener. In this space, the timbre changes can be reflected in the synthesized singing voice. The system was evaluated by the Euclidean distance in the space between an estimated result and a ground-truth under closed/open conditions.

Index Terms— Singing synthesis, Voice timbre changes, Singing information processing

1. INTRODUCTION

This paper describes a *singing-to-singing synthesis* system that automatically synthesizes a singing voice by mimicking a user's singing. Since 2007, many end users have started to use commercial singing synthesis systems to produce music and the number of listeners who enjoy synthesized singing is increasing. Over one-hundred-thousand copies of popular software packages[1] based on Vocaloid [1] have been sold and various compact discs that include synthesized vocal tracks have appeared on popular music charts in Japan. Singing synthesis systems are used not only to create original vocal tracks, but also for enjoying collaborative creation and communication via content-sharing services on the Web [2, 3]. Moreover, the systems are important tools for research into singing, since it can precisely control the pitch (fundamental frequency, F_0), dynamics (power), and voice timbre of a singing voice.

We previously developed a singing-to-singing synthesis system, *VocaListener*, that can estimate singing synthesis parameters of pitch and dynamics by mimicking a user's singing voice [4]. Since a natural voice is provided by the user, the synthesized singing voice mimicking it can be human-like and natural without time-consuming manual adjustment. Our aim in developing a system that can synthesize a singing voice without time-consuming manual adjustment is to help a user focus on "how to express the user's expression/message". Moreover, the ability to synthesize high-quality human-like singing voices will help us clarify the mechanisms of human singing voice production and perception. However, because VocaListener deals with only pitch and dynamics, it cannot express the overall expression of a user's singing.

A system which can reflect *voice timbre changes* of a user's singing voice in synthesized singing will be a useful tool for expanding the possibility of singing synthesis. There are many researches for manipulating voice timbre such as speaking voice conversion [5], emotional speech synthesis [6–8], and singing voice morphing [9], but they cannot deal with the timbre changes during singing. On the other hand, Vocaloid [1] enables a user to adjust singing synthesis parameters to manipulate acoustic features (e.g.,spectrum) of synthesized singing for each instant of time. The manual parameter adjustment is not easy, though, and requires considerable time and effort. Consequently, users tend to not change the parameters, or else change many parameters at the same time for each song or make only rough changes.

To deal with these problems, we propose *VocaListener2* that can synthesize a singing voice by mimicking the timbre changes in addition to the pitch and dynamics of the user's singing voice. Moreover, we propose an interface for adjusting the timbre changes to overcome limitations in the user's singing skills.

2. PREVIOUS SYSTEM AND ITS SHORTCOMING

To distinguish between our two systems, this paper refers to our previous VocaListener as *VocaListener1*. In this section, we describe the functions of VocaListener1 and its shortcoming, and discuss the problems we had to overcome to develop VocaListener2.

2.1. VocaListener1: A singing-to-singing synthesis system based on iterative parameter estimation

VocaListener1 [4] iteratively estimates parameters of pitch and dynamics[2] for a singing synthesis system (*e.g.,* Yamaha's Vocaloid [1]) so that the synthesized singing can become more similar to the user's singing (**Fig. 1**). The iterative estimation provides robustness with respect to different singing synthesis systems and their singer databases. The mean error values after the iteration are much smaller than with the previous approach [10] (see [4] for details)[3]. Moreover, VocaListener1 has a highly accurate lyrics-to-singing synchronization function, and its interface lets a user easily correct synchronization errors by simply pointing them out. In addition, VocaListener1 has a function to improve synthesized singing as if the user's singing skills were improved.

2.2. Extending the system to mimic voice timbre changes

To mimic timbre changes, we can take either of two approaches. One is the same as for VocaListener1 where we estimate parameters for singing synthesis software. However, we do not take this approach for two reasons:

[Reason 1] The voice timbre parameters depend on the system used: This approach is not robust with respect to different types of singing synthesis system because different systems have different parameters for manipulating the acoustic features of voice timbre. In fact, Vocaloid1 and Vocaloid2 [1] differ in some of the parameters used.

[Reason 2] An intermediate voice cannot be synthesized: For example, the *Hatsune Miku Append* singing synthesis software (referred to as MIKU Append)[4] can synthesize six kinds of voice (DARK,

This research was supported in part by CrestMuse, CREST, JST.

[1] http://www.vocaloid.com/product.html

[2] The estimated parameters are MIDI-based, such as the MIDI note number, pitch bend, pitch bend sensitivity, and expression.

[3] Demonstration videos including examples of synthesized singing are available at http://staff.aist.go.jp/t.nakano/VocaListener/

[4] http://www.crypton.co.jp/cv01a/ (in Japanese)

978-1-4577-0539-7/11/$26.00 ©2011 IEEE

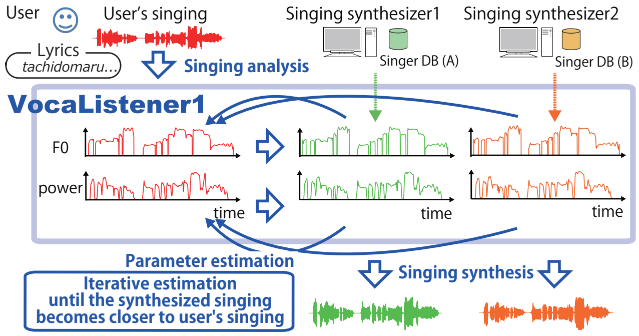

Fig. 1. Overview of VocaListener1, which iteratively estimates parameters of pitch and dynamics for singing synthesis from the user's singing voice and the song lyrics.

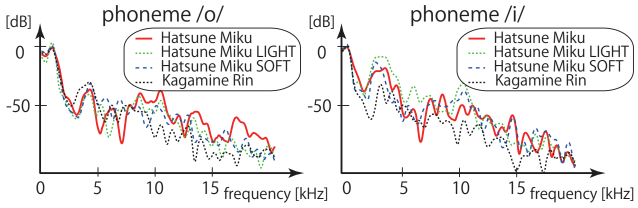

Fig. 2. Examples of differences in the spectral envelope due to phonetic and individual differences.

LIGHT, SOFT, SOLID, SWEET, and VIVID) which have the same individuality as with *Hatsune Miku* [11] and differ in voice timbre. However, it is difficult to synthesize an intermediate voice (*e.g.,* between LIGHT and SOLID) by using only the existing system.

The above limitations on mimicking a user's voice timbre changes based on the parameter estimation approach led us to take a novel approach based on signal processing. To develop a system able to mimic voice timbre changes, we need to solve two problems:
[Problem 1] How to represent voice timbre changes
[Problem 2] How to reflect timbre changes in a synthesized singing

Voice timbre changes can be defined as differences in the spectral envelope shape, such as the difference between Hatsune Miku and the MIKU Append. However, the spectral envelope depends on the phoneme (*e.g.,* between /o/ and /i/) and the individual (*e.g.,* between Hatsune Miku [11] and an another Vocaloid software[5]such as *Kagamine Rin*) as shown in **Fig. 2**. Therefore, timbre changes can be represented as a spectral envelope by suppressing such phonetic and individual effects. This made it possible to realize VocaListener2.

3. VOCALISTENER2: A SINGING SYNTHESIS SYSTEM MIMICKING VOICE TIMBRE CHANGES

We first use VocaListener1 to synthesize time-synchronized singing voices from several singer databases (DBs), and then estimate the spectral envelope of each sample of synthesized singing. The system constructs an M-dimensional *voice timbre space* by separating the voice timbre information and the phonetic information from the spectral envelopes of the singing voices using a subspace method. This is under the assumption that a space with large variance between singing samples is a voice timbre space because all singing voices and are synchronized for each time under pitch, dynamics, and phoneme. In this space, the singing voice is represented as a point in each time, and its temporal changes are represented as a trajectory. Such a subspace method can be used for speaker recognition by separating the phonetic space and the speaker space [12].

We then introduce an M-dimensional *timbre change tube*, which envelops the trajectories of the target timbre voices. In this paper, we suppose that the inside of this tube is a transposable area of the voice timbre. To get a spectral envelope reflecting the timbre changes in each time, a trajectory of the user's singing is adjusted so that it is inside the tube. A singing voice is then synthesized.

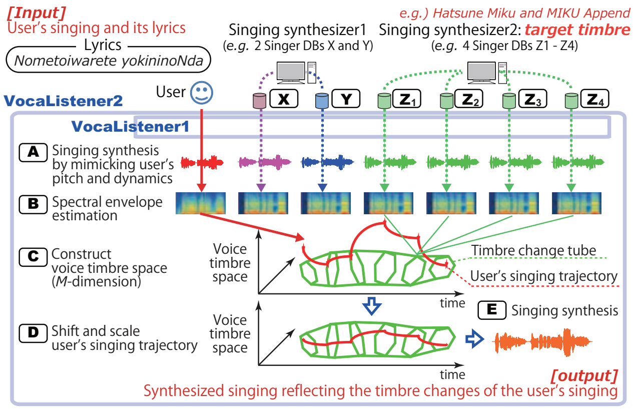
[Input]
User's singing and its lyrics
Lyrics
Nometoiwarete yokininoNda
e.g.) Hatsune Miku and MIKU Append
Singing synthesizer1
(e.g. 2 Singer DBs X and Y)
Singing synthesizer2: target timbre
(e.g. 4 Singer DBs Z1 - Z4)
VocaListener2
User
X
Y
Z1
Z2
Z3
Z4
VocaListener1
A Singing synthesis by mimicking user's pitch and dynamics
B Spectral envelope estimation
C Construct voice timbre space (M-dimension)
Voice timbre space
Timbre change tube
User's singing trajectory
time
E Singing synthesis
D Shift and scale user's singing trajectory
Voice timbre space
time
[output]
Synthesized singing reflecting the timbre changes of the user's singing

Fig. 3. Overview of VocaListener2, which automatically synthesizes a singing voice by mimicking pitch, dynamics, and timbre changes of a user's singing voice.

3.1. System outline

Figure 3 shows an overview of the VocaListener2 system. The system consists of VocaListener1, singing analysis ($\boxed{\text{A}}$–$\boxed{\text{D}}$), and singing synthesis ($\boxed{\text{E}}$). The user's singing voice and the lyrics are taken as the system input. The system synthesizes a singing voice that mimic the pitch, dynamics, and timbre changes of the user's singing by using singer DBs $\boxed{Z_1}$–$\boxed{Z_4}$ as the *target timbre*. These singer DBs, such as the Hatsune Miku and MIKU Append, have different voice timbres while keeping the same individuality. Throughout this paper, singing samples are monaural recordings of solo vocal digitized at 16 bit/44.1 kHz.

Using the input, the system first uses VocaListener1 to automatically synthesize time-synchronized singing voices from several singer DBs ($\boxed{\text{A}}$). Second, the system estimates the spectral envelope of each sample of synthesized singing ($\boxed{\text{B}}$), since this envelope represents the voice timbre and is independent of the F_0. The system then constructs the M-dimensional voice timbre space ($\boxed{\text{C}}$) by a subspace method. A spectral envelope reflecting the timbre changes is estimated from an adjusted trajectory of the user's singing so that it is inside the tube ($\boxed{\text{D}}$). Finally, a singing voice is synthesized from the spectral envelopes ($\boxed{\text{E}}$).

3.2. Singing analysis

The system estimates the spectral envelope and the voice timbre space, and adjusts the trajectory of the user's singing as follows. Since the analysis frame is shifted by 44.1 samples, the discrete time step (1 *frame-time*) is 1 ms. This paper uses time t for the time measured in frame-time units.

Spectral envelope estimation $\boxed{\text{B}}$: The spectral envelope is estimated using the STRAIGHT speech manipulation system [13]. The estimated spectral envelope is represented by 2045 bins, and converts to 80th order DCT (discrete cosine transform) coefficients.

Voice timbre space construction $\boxed{\text{C}}$: **Figure 4** shows the estimation process. The system first applies PCA (principal components analysis) for *each* voiced frame between singing voices, and low $N(t)$-dimensional features can be estimated for each frame. The $N(t)$-dimension is decided under the cumulative contribution

[5] http://www.vocaloid.com/product.html

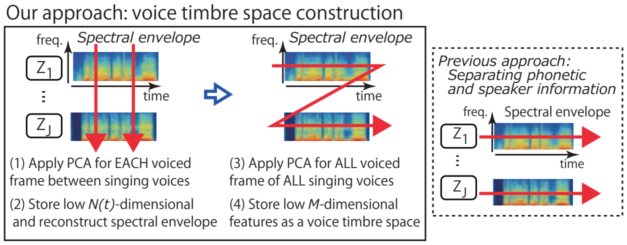
Our approach: voice timbre space construction
freq.
Spectral envelope
Z_1
Z_J
time
freq.
Spectral envelope
time
(1) Apply PCA for EACH voiced frame between singing voices
(2) Store low $N(t)$-dimensional and reconstruct spectral envelope
(3) Apply PCA for ALL voiced frame of ALL singing voices
(4) Store low M-dimensional features as a voice timbre space
Previous approach: Separating phonetic and speaker information
freq.
Spectral envelope
Z_1
Z_J
time

Fig. 4. Voice timbre space estimation and previous subspace method for speaker recognition.

ratio ($\geq$ 80%). However, this means that different spaces are constructed between each pair of frames. The system therefore reconstruct spectral envelopes from the $N(t)$-dimensional features, applies PCA again for *all* voiced frame of *all* singing voices, and stores low M-dimensional features ($M = 3$). This 3-dimensional space is used as a voice timbre space.

Adjustment of the user's singing trajectory $\boxed{\text{D}}$: The user's singing and the voices of the target timbre in the voice timbre space are shifted and scaled to 0 to 1 for each dimension, respectively.

3.3. Singing synthesis

To achieve a singing voice that mimics the timbre changes of the user's voice, the singing voice is synthesized from the trajectory in the voice timbre space $\boxed{\text{E}}$. **Figure 5** shows an example of placement for the user's singing and target timbre voices which are Hatsune Miku and MIKU Append (DARK, LIGHT, SOFT, SOLID, SWEET, and VIVID)[6]. Each spectral envelope is represented as a point in the space for each frame-time.

The problem is to estimate a spectral envelope from the placement and those spectral envelopes. To estimate such a spectral envelope, we first introduce a *spectral transform curve* which is difference between the spectral envelope of a standard voice (*e.g.,* Hatsune Miku) and the others for each frame. All spectral transform curves for all frames are referred as a *spectral transform surface*. Let $Zr_j(f,t)$ be a spectral transform surface for frequency f and time t defined as follows, where $Z_{j=1,2,\cdots,J}(f,t)$ is the jth spectral envelope[7] ($j = 1$ indicates the standard voice),

$$Zr_j(f,t) = \log\left(\frac{Z_j(f,t)}{Z_1(f,t)}\right). \tag{1}$$

Using these, the spectral transform surface mimicking the timbre changes $g(\mathbf{u}(t); f, t)$ is estimated by applying a *variational interpolation* method based on radial basis function [14] as follows. We solve for the set of $w_j(f,t)$ that will satisfy the interpolation constraints as equation (4).

$$g(\mathbf{u}(t); f,t) = \sum_{k=1}^{J}\left(w_k(f,t)\cdot\phi\left(\mathbf{u}(t) - \mathbf{z}_k(t)\right)\right) + P(\mathbf{u}(t); f,t), \tag{2}$$

$$Zr_j(f,t) = \sum_{k=1}^{J}\left(w_k(f,t)\cdot\phi\left(\mathbf{z}_j(t) - \mathbf{z}_k(t)\right)\right) + P(\mathbf{z}_j(t); f,t), \tag{3}$$

$$g(\mathbf{z}_j(t); f,t) = Zr_j(f,t), \tag{4}$$

$$P(\mathbf{x}; f,t) = p_0(f,t) + \sum_{m=1}^{M} p_m(f,t)\cdot x^{(m)}, \tag{5}$$

where $\mathbf{u}(t)$ indicates a point of the trajectory of the user's singing, $\mathbf{z}_j(t)$ are the locations of the constraints (*i.e.* the target timbre voices), $w_j(f,t)$ are the weights, $\phi(\cdot)$ indicates a distance between vectors (in this paper, $\phi(\cdot) = |\cdot|$), and $P(\cdot; f,t)$ is a degree one polynomial in M variables. Since the above equation is linear with respect to the unknowns, $w_j(f,t)$, and the coefficients of $P(\cdot; f,t)$, it can be formulated as a linear system for each time t and written as:

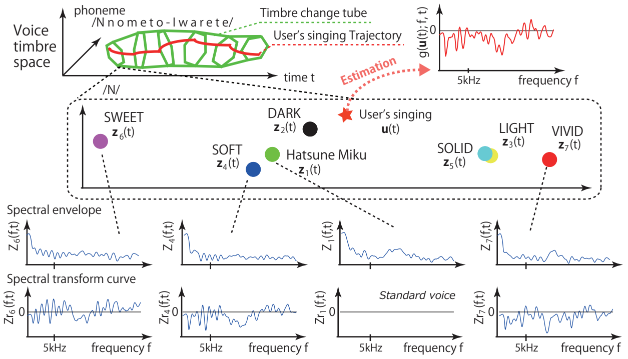

Fig. 5. Example of an estimated spectral envelope from vectors for user's singing and target timbre voices (Hatsune Miku and MIKU Append) in the voice timbre space.

$$\begin{bmatrix} \phi_{11} & \cdots & \phi_{1J} & 1 & z_1^{(1)} & z_1^{(2)} & z_1^{(3)} \\ \phi_{21} & \cdots & \phi_{2J} & 1 & z_2^{(1)} & z_2^{(2)} & z_2^{(3)} \\ \vdots & & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_{J1} & \cdots & \phi_{JJ} & 1 & z_J^{(1)} & z_J^{(2)} & z_J^{(3)} \\ 1 & \cdots & 1 & 0 & 0 & 0 & 0 \\ z_1^{(1)} & \cdots & z_J^{(1)} & 0 & 0 & 0 & 0 \\ z_1^{(2)} & \cdots & z_J^{(2)} & 0 & 0 & 0 & 0 \\ z_1^{(3)} & \cdots & z_J^{(3)} & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_J \\ p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = \begin{bmatrix} Zr_1 \\ Zr_2 \\ \vdots \\ Zr_J \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6}$$

where ϕ_{ij} indicates $\phi(\mathbf{z}_i(t) - \mathbf{z}_j(t))$, and f and t are omitted for this description.

The estimated spectral transform surface is then thresholded to reduce the unnaturalness of synthesis. Moreover, to preserve the continuity of spectral envelopes, a time-frequency smoothing FIR filter is applied the surface. Finally, the surface is applied to the standard spectral envelope, and then the singing voice is synthesized by using STRAIGHT.

3.4. An interface for adjusting the timbre changes

To extend the flexibility and to overcome the limitation of the user's singing ability, we propose an interface which has three important functions:
Scaling function: to emphasize/suppress voice timbre fluctuations.
Shifting function: to synthesize around a particular voice timbre, the center of the voice timbre fluctuations can be changed.
Scaling/shifting in part: to adjust in detail by partially, applying above two functions.

4. EXPERIMENTAL EVALUATION

VocaListener2 was tested in two experiments. In these experiments, we used 17 singer DBss (3 male and 14 female voices) for synthesizing Japanese singing for two commercial singing synthesis software programs based on Yamaha's Vocaloid or Vocaloid2 technology [1]. Seven of the DBs, Hatsune Miku and MIKU Append, were used for the target timbre. An unaccompanied song sample (solo vocal) was taken from the RWC Music Database (Music Genre [15]:

[6] The timbre change tube shown on the upper left is an image

[7] j indicates the number of the voice of the target timbre, and J is 7 (one is Hatsune Miku and six is MIKU Append) in this paper.

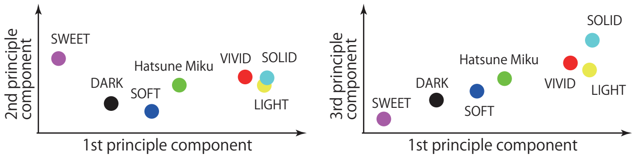

Fig. 6. Constructed 3-dimensional voice timbre space and average vectors of first three principle components for each voice timbre.

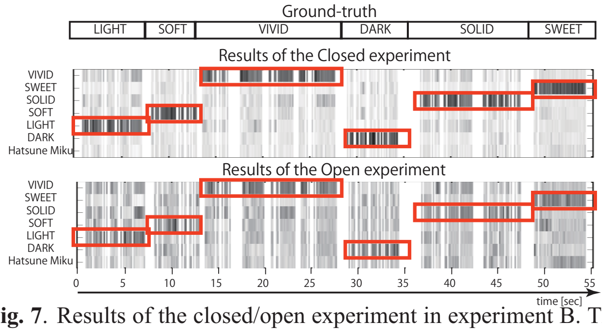

Fig. 7. Results of the closed/open experiment in experiment B. The Euclidean distance (the nearer to the timbre, the darker in the color) between each voice timbre and the estimated trajectory in the voice timbre space are shown.

RWC-MDB-G-2001 No.91), and was used as the user's singing. Since the user's singing sung by male singer, the female singing was synthesized at an octave higher by applying the pitch transpose (VocaListener-plus) function [4].

4.1. Experiment A: voice timbre space construction

To evaluate the property of the constructed voice timbre space, experiment A used a sample of the user's singing that was 55 s in length. **Figure 6** shows the constructed 3-dimensional voice timbre space and average vectors of the first three principle components of each voice timbre. The average of the $N(t)$ is 4.18, and the cumulative contribution ratio of the space was 42.3%. The results of Fig. 6 suggest that a low 3-dimensional space can separate each voice timbre. In addition, the positional relationship of each average vector qualitatively reflects the auditory impression. These findings suggest our system can construct an appropriate voice timbre space.

4.2. Experiment B: mimicking voice timbre changes

Two synthesized singing voices by using VocaListener1 from MIKU Append as singer DBs were used as an input, which consisted of six kinds of voice timbre. **Figure 7** shows its ground-truth and the Euclidean distance between voice timbres and the estimated user's trajectory in the voice timbre space under closed/open conditions. To evaluate results under a *closed condition*, an input was synthesized from the same voice timbre used to construct the voice timbre space. To evaluate results under an *open condition*, an input was synthesized from a different voice timbre (by changing the synthesized parameter GEN for Vocaloid2 to 90 from the default of 64).

The distance under closed condition was approximately correct. However, there were some errors under open condition. Most of the false estimations were due to neighbors such as LIGHT, SOLID, and VIVID (see Fig. 6). The proximity of the neighbors in the space suggests the neighbors have a similar spectral envelope. Since our system estimates the spectral envelope using all of the target timbres weighted by distance by eq. (2), the estimated result has less influence for the false estimations. This suggests that VocaListener2 is an effective way to mimic the timbre changes of a user's singing voice.

5. CONCLUSION AND FUTURE DIRECTION

To develop a system which can synthesize a singing voice by mimicking the timbre changes of a user's singing voice, we introduce a *voice timbre space* and a *timbre change tube* as a novel signal processing approach where we apply variational interpolation. Experimental results suggest that the system effectively mimics target singing. In our experience of synthesizing a song with VocaListener2 using Hatsune Miku and MIKU Append, we found the synthesized quality was high[8]. With VocaListener2, a singing voice mimicking pitch, dynamics, and various voice timbres can be easily synthesized. VocaListener2 therefore promises to become fundamental tools for research into singing. For example, our goal is to use VocaListener2 to clarify the mechanism of human singing voice production and perception.

One benefit of VocaListener2 is that a user does not need to perform time-consuming manual adjustment. Moreover, the VocaListener2 framework is scalable by changing the estimated spectral envelope. For example, by applying the spectral transform surface for Hatsune Miku to a spectral envelope from Kagamine Rin as another singer DB, we should be able to achieve a spectral envelope for *pseudo* Kagamine Rin Append. However, this may be an oversimplification and we will investigate whether this can be done in our future work.

6. REFERENCES

[1] H. Kenmochi *et al.*, "Vocaloid – commercial singing synthesizer based on sample concatenation," in *Proc. INTERSPEECH 2007*, 2007, pp. 4011–4010.

[2] M. Hamasaki *et al.*, "Network analysis of massively collaborative creation of multimedia contents: Case study of hatsune miku videos on nico nico douga," in *Proc. uxTV' 08*, 2008, pp. 165–168.

[3] Cabinet Office, Government of Japan, "Virtual idol," in *Highlighting JAPAN through images*, 2009, vol. 2, pp. 24–25, http://www.gov-online.go.jp/pdf/hlj_img/vol_0020et/24-25.pdf.

[4] T. Nakano *et al.*, "VocaListener: A singing-to-singing synthesis system based on iterative parameter estimation," in *Proc. SMC 2009*, 2009, pp. 343–348.

[5] T. Toda *et al.*, "Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[6] O. Türk *et al.*, "A comparison of voice conversion methods for transforming voice quality in emotional speech synthesis," in *Proc. Interspeech 2008*, 2008, pp. 2282–2285.

[7] T. Nose *et al.*, "Hmm-based style control for expressive speech synthesis with arbitrary speaker's voice using model adaptation," *IEICE Trans. on Information and Systems*, vol. E92-D, no. 3, pp. 489–497, 2009.

[8] Z. Inanoglua *et al.*, "Data-driven emotion conversion in spoken English," *Speech Communication*, vol. 51, Is. 3, pp. 268–283, 2009.

[9] H. Kawahara *et al.*, "Temporally variable multi-aspect auditory morphing enabling extrapolation without objective and perceptual breakdown," in *Proc. ICASSP2009*, 2009, pp. 3905–3908.

[10] J. Janer *et al.*, "Performance-driven control for sample-based singing voice synthesis," in *Proc. DAFx-06*, 2006, pp. 41–44.

[11] Crypton Future Media, "What is the hatsune miku movement?," http://www.crypton.co.jp/download/pdf/info_miku_e.pdf.

[12] M. Nishida *et al.*, "Speaker recognition by separating phonetic space and speaker space," in *Proc. EUROSPEECH2001*, 2001, pp. 1381–1384.

[13] H. Kawahara *et al.*, "Restructuring speech representations using a pitch adaptive time-frequency smoothing and an instantaneous frequency based on F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, vol. 27, pp. 187–207, 1999.

[14] G. Turk *et al.*, "Modelling with implicit surfaces that interpolate," *ACM Trans. on Graphics*, vol. 21, no. 4, pp. 855–873, 2002.

[15] M. Goto *et al.*, "Rwc music database: Music genre database and musical instrument sound database," in *Proc. ISMIR2003*, 2003, pp. 229–230.

[8] Demonstration videos including examples of synthesized singing are available at http://staff.aist.go.jp/t.nakano/VocaListener2/.