ISCA Archive
http://www.isca-speech.org/archive

7th ISCA Workshop on
Speech Synthesis (SSW-7)
Kyoto, Japan
September 22-24, 2010

Recent Development of the HMM-based Singing Voice Synthesis System — Sinsy

Keiichiro Oura, Ayami Mase, Tomohiko Yamada, Satoru Muto, Yoshihiko Nankaku, and Keiichi Tokuda

Department of Computer Science, Nagoya Institute of Technology, Japan
{uratec,ayami-m,piko3141,mutest,nankaku}@sp.nitech.ac.jp, tokuda@nitech.ac.jp

Abstract

A statistical parametric approach to singing voice synthesis based on hidden Markov Models (HMMs) has been grown over the last few years. The spectrum, excitation, and duration of singing voices in this approach are simultaneously modeled with context-dependent HMMs and waveforms are generated from the HMMs themselves. In December 2009, we started a free on-line singing voice synthesis service called "Sinsy." Users can obtain synthesized singing voices by uploading musical scores represented in MusicXML to the Sinsy website. The present paper describes recent developments of Sinsy in detail.

Index Terms: HMM-based speech synthesis, singing voice synthesis

1. Introduction

A statistical parametric approach to speech synthesis based on hidden Markov models (HMMs) has grown in popularity over the last few years [1]. Context-dependent HMMs are estimated from speech databases in this approach, and speech waveforms are generated from the HMMs themselves. This framework makes it possible to model different voice characteristics, speaking styles, or emotions without recording large speech databases. For example, adaptation [2], interpolation [3], and eigenvoice techniques [4] have been applied to this system, which demonstrated that voice characteristics could be modified. A singing voice synthesis system has also been proposed by applying the HMM-based approach [5]. In December 2009, we publicly released a free on-line singing voice synthesis service called "Sinsy (HMM-based Singing Voice Synthesis System)" [6]. One of features of the system is that it was constructed using open-source software packages, e.g., HTS [7], hts_engine API [8], SPTK [9], STRAIGHT [10], and the Crest-MuseXML Toolkit [11]. Users can synthesize singing voices by uploading musical scores represented in MusicXML [12] to the website. To construct the system, we have introduced three specific techniques, i.e., a new definition of rich contexts, vibrato modeling, and a pruning approach using note boundaries. The present paper describes these recent developments of Sinsy in detail.

The rest of this paper is organized as follows. Section 2 gives an overview of the HMM-based singing voice synthesis system. Section 3 describes techniques that have been proposed for training. Details of Sinsy are presented in Section 4. Concluding remarks are made in Section 5.

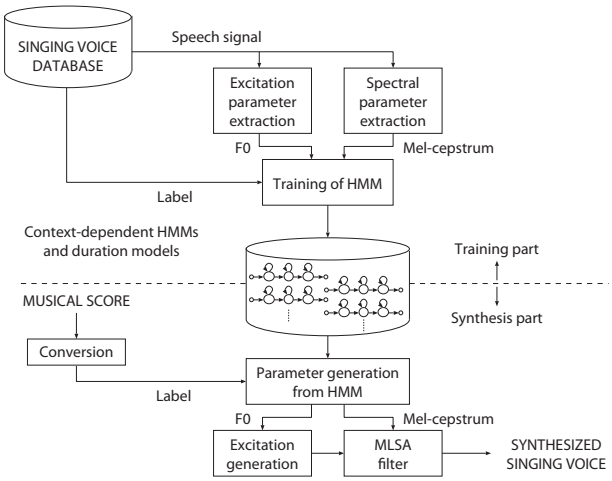


Figure 1: *Overview of HMM-based singing voice synthesis system.*

2. HMM-based singing voice synthesis system

The HMM-based singing voice synthesis system is quite similar to the HMM-based text-to-speech synthesis system [1]. However, there are distinct differences between them. This section overviews the baseline singing voice synthesis system and then gives details of the differences between the HMM-based text-to-speech synthesis and the baseline singing voice synthesis systems.

2.1. System overview

Figure 1 gives an overview of the HMM-based singing voice synthesis system [5]. It consists of training and synthesis parts. The spectrum (e.g., mel-cepstral coefficients [13]) and excitation (e.g., fundamental frequencies: F_0s) in the training part are extracted from a singing voice database and they are then modeled with using context-dependent HMMs. Context-dependent models of state durations are also estimated. An arbitrarily given musical score including the lyrics to be synthesized is first converted in the synthesis part to a context-dependent label sequence. Second, according to the label sequence, an HMM corresponding to the song is constructed by concatenating the context-dependent HMMs. Third, the state durations of the song HMM are determined with respect to the state duration models. Fourth, the spectrum and excitation parameters are generated by the speech parameter generation algorithm [14]. Finally, a singing voice is synthesized directly from the gener-

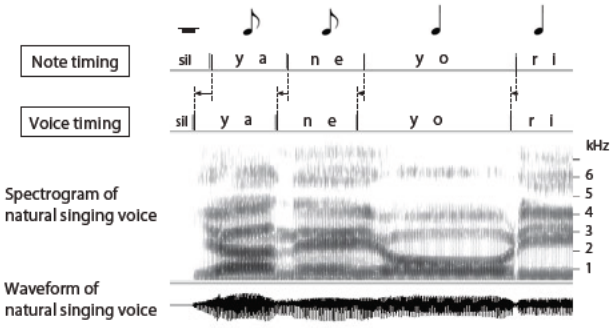


Figure 2: *Example of timing.*

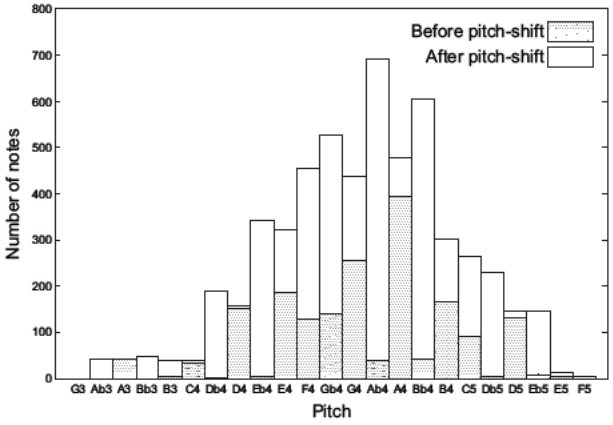


Figure 3: *Distribution of pitch in training data (10 songs).*

ated spectrum and excitation parameters by using a Mel Log Spectrum Approximation (MLSA) filter [15].

2.2. Timing model

One of the unique features of the singing voice synthesis system is the timing model [5]. In singing voice synthesis, the rhythm or tempo of the music must not be ignored when the singing voice is synthesized. Therefore, the start of timing of the notes or phoneme durations in each note must be determined according to the musical score. However, there are differences between the start of timing of the notes and singing voices, as shown in Figure 2. The start of timing of the singing voice is earlier than that of a corresponding note.

The timing of each note is modeled with Gaussian distributions to overcome this problem. Hidden semi-Markov model (HSMM) [16]-based phoneme alignments performed by using weighted finite-state transducers (WFST) [17] are used to find the start of timing of each note in the singing voice database. The timing models are then trained as context-dependent models, and decision-tree based context-clustering is applied to them in the same manner as the other models in the system. The timing models control the head of each note, which generally consists of more than one phoneme. The length of each note is determined by the timing model in the synthesis part as:

$$T_k' = T_k - g_{k-1} + g_k, \tag{1}$$

where T_k, T_k', and g_k correspond to the length of the k-th note on the musical score, the length of the k-th note after the timing has been applied, and the timing of the k-th note determined by the timing models.

2.3. Pitch-shifted pseudo-data

HMM-based speech synthesis systems heavily depend on the training data in performance because these systems are "corpus-based." Therefore, HMMs corresponding to contextual factors that hardly ever appear in the training data cannot be well-trained. Algorithms for designing speech databases taking into consideration the balance among contextual factors have been proposed [18] to solve this problem. Databases including various contextual factors should also be used in HMM-based singing voice synthesis systems. However, data have to be sparse because singing voices have numerous contextual factors, e.g., pitch, tempo, key, beat, and dynamics, in addition to those used in reading speech synthesis. Pitch should especially be correctly covered since generated F_0 trajectories have a great impact on the subjective quality of synthesized singing voices. Therefore, we applied a technique to training HMMs by using pitch-shifted pseudo-data [19]. Since pitch is represented by a log F_0 parameter, pitch-shifted pseudo-data can easily be prepared by shifting log F_0 up or down in halftones. This technique makes it possible to increase the amount of F_0 training data without recording large amounts of singing voice data.

Figure 3 shows the distributions of pitch in part of the training data (10 songs). The distribution has a problem with sparseness before pitch-shifted is applied. As the amount of data increases, the distribution smoothes after pitch-shifted pseudo-data are added. Mel-cepstral coefficients were added by copying the same data since we assumed that they were not affected by the small amount of pitch-shifting.

The amount of training data is increased threefold by adding pitch-shifted pseudo-data. Therefore, decision trees increase in size when the minimum description length (MDL) based criterion [20] is used. Thus, context-clustering should be stopped when each decision tree reaches an appropriate size. The MDL criterion is used in the HMM-based singing voice synthesis system to determine when to stop splitting nodes. The change in total description lengths before and after splitting is calculated, and splitting is conducted when the difference exceeds a threshold. This node splitting is carried out until no nodes exceed the threshold. Therefore, the number of leaf nodes in each decision tree increases if the threshold is set lower, and reduces if the threshold is set higher. When node S is divided into two nodes, S_{q+} and S_{q-}, by question q, the change of total description lengths with this split is calculated as:

$$\Delta_q = \mathcal{L}(S) - \left\{\mathcal{L}\left(S_{q+}\right) + \mathcal{L}\left(S_{q-}\right)\right\} + \alpha\frac{N}{2}\log\Gamma(S_0), \tag{2}$$

where S_0 denotes a root node, α is a heuristic weight[1] for the penalty term of the MDL criterion, and N is the number of parameters increased by this split. Here, $\Gamma(\cdot)$ is the posterior probability. The heuristic weight, α, is used to control the size of the decision trees.

3. Proposed techniques for training

We propose three specific techniques for singing voice synthesis to make singing voices more natural, i.e., a new definition of rich contexts, vibrato modeling, and a pruning approach using note boundaries. The following three subsections provide details of these.

[1]The standard value of α is unity in the MDL criterion.

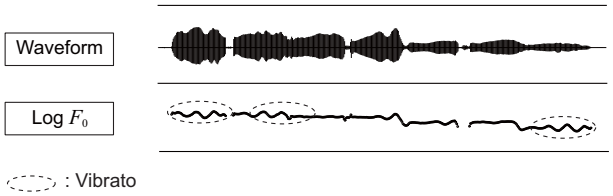


Figure 4: *Example of vibrato parts in F_0 sequence.*

3.1. Definition of rich contexts

Contextual factors that may affect reading speech, e.g., phoneme identity, parts-of-speech, accent, and stress, have been taken into account [1] in the HMM-based text-to-speech synthesis system. However, the contextual factors that affect the singing voice should differ from those used in text-to-speech synthesis. We redesigned rich contexts for the HMM-based singing voice synthesis discussed in this paper. The following contextual factors were considered for Sinsy:

- Phoneme
  - Quinphone: a phoneme within the context of two immediately preceding and succeeding phonemes.
- Mora[2]
  - The number of phonemes in the {previous, current, next} mora.
  - The position of the {previous, current, next} mora in the note.
- Note
  - The musical tone, key, beat, tempo, length, and dynamics of the {previous, current, next} note.
  - The position of the current note in the current measure and phrase.
  - The tied and slurred flag.
  - The distance between the current note and the {next, previous} accent and staccato.
  - The position of the current note in the current crescendo and decrescendo.
- Phrase
  - The number of phonemes and moras in the {previous, current, next} phrase.
- Song
  - The number of phonemes, moras, and phases in the song.

These contexts can automatically be determined from the musical score including the lyrics. We covered those contexts that were considered necessary to organize hierarchy and symmetry.

3.2. Vibrato model

Vibrato is one of the important singing techniques that should be modeled, even though it is not included in the musical score. Figure 4 shows examples of vibrato parts in an F_0 sequence. The timing and intensity of vibrato vary from singer to singer. Therefore, vibrato modeling is required to make the synthesized singing voice mora natural. However, small fluctuations such as vibrato are smoothed through the HMM training and synthesis process in the HMM-based singing voice synthesis system.

[2]The Japanese mora is a sound unit consisting of either one or two phonemes.

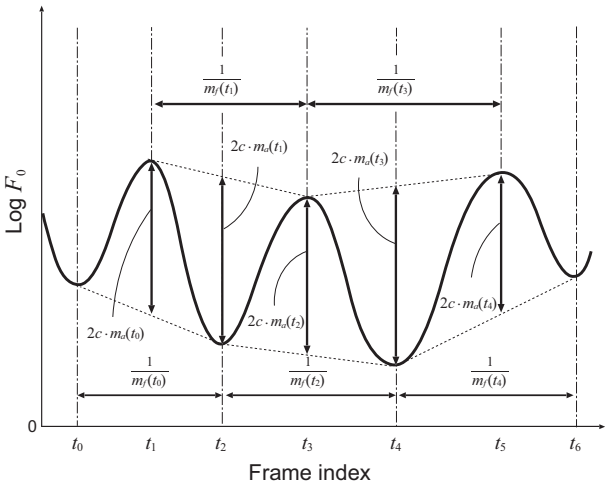


Figure 5: *Analysis of vibrato parameters.*

We introduced a simple vibrato modeling technique for HMM-based singing voice synthesis [21] to model vibrato automatically.

Vibrato has been assumed as periodic fluctuations of only F_0 for the sake of simplicity in this paper. The vibrato, $v(\cdot)$, of the t frame can be defined as

$$v\left(m_a(t), m_f(t), i\right) = m_a(t) \sin\left(2\pi\, m_f(t)\, f_s\, (t - t_0)\right), \qquad (3)$$

where $m_a(t)$, $m_f(t)$, and f_s correspond to the F_0 amplitude of vibrato in cents, the F_0 frequency of vibrato in Hz, and frame shift. Two parameters, amplitude in cents and frequency in Hz, are used for training and synthesis.

Vibrato sections are estimated from a log F_0 sequence [22]. Restrictions of amplitude and frequency are based on previous research [23, 24] with an amplitude range from 30 to 150 cents and a frequency range from 5 to 8 Hz. Figure 5 shows the analysis of vibrato amplitude and frequency. Note that c is defined as $\log 2/1200$ for conversion from cents to log Hz.

Two dimensional vibrato parameters, m_a and m_f, are added to the observation vector in the training part. When each observation vector $\boldsymbol{o}_t$ consists of spectrum $\boldsymbol{o}_t^{(spec)}$, excitation $\boldsymbol{o}_t^{(F_0)}$, and vibrato $\boldsymbol{o}_t^{(vib)}$, the state output probability, $b_s(\boldsymbol{o}_t)$, of the s-th state is given by

$$b_s(\boldsymbol{o}_t) = p_s^{\gamma_{spec}}\left(\boldsymbol{o}_t^{(spec)}\right) \cdot p_s^{\gamma_{F_0}}\left(\boldsymbol{o}_t^{(F_0)}\right) \cdot p_s^{\gamma_{vib}}\left(\boldsymbol{o}_t^{(vib)}\right) \qquad (4)$$

where γ_{spec}, γ_{F_0}, and γ_{vib} correspond to the heuristic weights for the spectrum, excitation, and vibrato.

3.3. Pruning approach using note boundaries

The computational cost is expensive to train HMM-based singing voice synthesis systems because singing voices are longer than normal utterances. HMMs are usually trained based on the EM algorithm with the maximum likelihood (ML) criterion [1]. When a state sequence is determined, the joint probability of an observation vector sequence and a state sequence is calculated by multiplying the state transition probabilities and the output probabilities for each state. Because this calculation is computationally expensive, the forward-backward algorithm and the pruning approach are generally used to reduce the computational cost. However, estimating the optimal state sequence

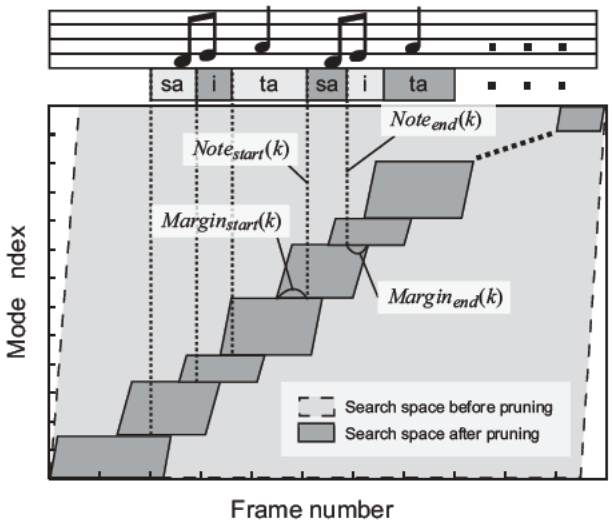


Figure 6: *Pruning approach using note boundaries.*

can fail with the pruning approach. This paper proposes a pruning approach using note boundaries [25] (Figure 6). Since the timing between notes and singing voices is not large, the state, i, at time t can be restricted by note boundaries. The only states corresponding to the current note in the interval from

$$t = Note_{start}(k) - Margin_{start}(k) \quad (5)$$

to

$$t = Note_{end}(k) + Margin_{end}(k) \quad (6)$$

should be searched, where k denotes the note index. Thus, the technique restricts the number of failures that cannot estimate the optimal state sequence, and the amount of computational time can be reduced.

4. Tools

One of the characteristics of Sinsy is that it consists of open-source software. All software used in each module is described.

4.1. Speech Signal Processing Toolkit (SPTK)

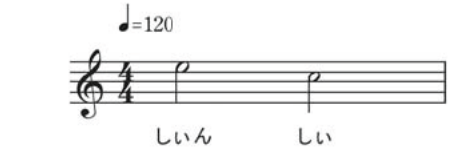
The speech signal processing toolkit (SPTK-3.3) [9] is a suite of speech signal processing tools for UNIX environments. This software is released under a new and simplified Berkeley Software Distribution (BSD) license [26]. The SPTK commands were used for mel-cepstral analysis [13] and composing the training data.

4.2. STRAIGHT

STRAIGHT V40 [27, 10] is a tool for manipulating timbre and pitch. It is continuously being evolved to attain better sound quality, which is closer to original natural speech, through the introduction of advanced signal-processing algorithms. In Sinsy, STRAIGHT is used to obtain speech spectra. Note that the license for STRAIGHT is different from that for SPTK.

4.3. HMM-based Speech Synthesis System (HTS)

The HMM-based speech synthesis system (HTS-2.1.1) [28, 7], which provides a research and development platform for statistical parametric speech synthesis is used for HMM training.




```
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE score-partwise PUBLIC
"-//Recordare//DTD MusicXML 2.0 Partwise//EN"
"http://www.musicxml.org/dtds/partwise.dtd">
...
<measure number="1">
 <attributes>
  <divisions>1</divisions>
  <key>
   <fifths>0</fifths>
   <fifths>major</fifths>
  </key>
  <time>
   <beats>4</beats>
   <beat-type>4</beat-type>
  </time>
...
 </attributes>
 <sound tempo="120"/>
 <note>
  <pitch>
   <step>E</step>
   <octave>5</octave>
  </pitch>
  <duration>2</duration>
  <type>half</type>
  <lyric>
   <text>・・・・</text>
  </lyric>
 </note>
 <note>
  <pitch>
   <step>C</step>
   <octave>5</octave>
  </pitch>
  <duration>2</duration>
  <type>half</type>
  <lyric>
   <text>・・</text>
  </lyric>
 </note>
</measure>
</part>
</score-partwise>
```

Figure 7: *Example of MusicXML.*

Various organizations currently use it to conduct their own research projects. HTS has been developed by the HTS working group as an extension of the HMM Toolkit (HTK) [29]. The source code for HTS has been released as an HTK patch. Although the patch has been released under the new and simplified BSD license [26], once the patch is applied users must comply with the HTK license.[3] The HTS patch code, associated files, and scripts for the HMM-based singing voice synthesis system can be downloaded from the HTS website.

4.4. CrestMuseXML Toolkit (CMX)

MusicXML 2.0 [12], which represents the musical score, is used for Sinsy to load the pitch, lyric, tempo, key, beat, dynamics, etc. Figure 7 has an example of MusicXML. MusicXML has been developed by Recordare to create an Internet-friendly

[3]The HTK license prohibits redistribution and commercial use of source, object, or executable codes.

method of publishing musical scores. CMX-0.50 [30, 11], which can analyze MusicXML, is used for the front-end of the synthesis part.

4.5. HMM-based Speech Synthesis Engine (hts_engine API)

A small stand-alone run-time synthesis engine called hts_engine API-1.03 [8] is used for the back-end of the synthesis part. It works without the HTK (HTS) libraries, and it has been released under the new and simplified BSD license [26] on the SourceForge site. Users can develop their own open or proprietary software based on the run-time synthesis engine, and redistribute these source, object, and executable codes without any restrictions.

5. Details of Sinsy

5.1. Training conditions

Seventy children's songs (total: 70 min) by female singer f001 were used for training. Singing voice signals were sampled at 48kHz and windowed with a 5-ms shift, and mel-cepstral coefficients [13] were obtained from STRAIGHT spectra [27]. The feature vectors consisted of spectrum, excitation, and vibrato parameters. The spectrum parameter vectors consisted of 49 STRAIGHT mel-cepstral coefficients including the zero coefficient, their delta, and delta-delta coefficients. The excitation parameter vectors consisted of log F_0, its delta, and delta-delta. The vibrato parameter vectors consisted of amplitude (cent) and frequency (Hz), their delta, and delta-delta coefficients. The range of pitch-shifted pseudo data was ± a halftone.

A seven-state (including the beginning and ending null states), left-to-right, no-skip structure was used for the HSMM [16]. The spectrum stream was modeled with single multivariate Gaussian distributions. The excitation stream was modeled with multi-space probability distributions HSMM (MSD-HSMM) [31], each of which consisted of a Gaussian distribution for "voiced" frames and a discrete distribution for "unvoiced" frames. The vibrato stream was also modeled with MSD-HSMMs, each of which consisted of a Gaussian distribution for "vibrato" frames and a discrete distribution for "unvibrato" frames. The state durations of each model were modeled with a five-dimensional (equal to the number of emitting states in each model) multi-variate Gaussian distribution. The heuristic weights for the spectrum, F_0, and vibrato in Equation (4) were set to 1.0, 1.0, and 0.0. The decision tree-based context-clustering technique was separately applied to distributions for the spectrum, excitation, vibrato, state duration, and timing. The MDL criterion [20] was used to control the size of the decision trees. The heuristic weight, α, for the penalty term in Equation (2) was 5.0. Although the decision tree-based context-clustering technique was separately applied to distributions for the spectrum, excitation, vibrato, state duration, and timing, the same α was used. To obtain a natural synthetic singing voice, minimum generation error (MGE) training with the Euclidean distance [32] was applied to the spectrum, excitation, and vibrato stream after ML-based HSMM training. A speech parameter generation algorithm taking into consideration context-dependent global variance (GV) without silence [33] was used for generating the parameters.

The number of leaf nodes in the decision trees is listed in Table 1. Table 2 lists the total file sizes for Sinsy. The total file size for Sinsy is no more than 2.5 MBytes with 48 kHz sampling-rate.

Table 1: *Number of leaf nodes in decision trees.*

Mel-cepstrum	648
F_0	1489
Vibrato	1684
State duration	144
Timing	114

Table 2: *The total file sizes for Sinsy (KBytes).*

Front-end program (CMX)	456
Phoneme table	3
Back-end program (hts_engine API)	677
Acoustic model	1652
Total file size for Sinsy	2588

5.2. On-line service

A web-based user interface [6] was adopted for Sinsy (Figure 8). One of the reasons for this was that Sinsy could be frequently updated. Users can easily change the timbre, pitch, and strength of the vibrato. The website placed some restrictions on the use of Sinsy. The first restriction was the range of pitches, because a pitch that hardly ever appeared in the training data could not be synthesized in the HMM-based singing voice synthesis system. Therefore, MusicXML files that exceeded the range of pitches from G3 to F5 were rejected. The second restriction was the length of the synthesized singing voice. One of the most attractive features of HMM-based singing voice synthesis is its small computational cost in the synthesis part. However, this system is vulnerable to frequent access or long songs because singing voices are synthesized on the web server. Therefore, MusicXML files that exceed 5 min are rejected. The rate at which waveforms were properly synthesized by utilizing user's MusicXML files that were uploaded to Sinsy from January to April 2010 was about 70 %. The other 30 % included error, other than that created by these restrictions, that could not convert MusicXML files because of the differences in MusicXML files generated by various tools.

6. Conclusions

This paper described recent developments in the HMM-based singing voice synthesis system (Sinsy). To obtain natural singing voices, we proposed three specific techniques for singing voice synthesis: the definition of rich contexts, the vibrato model, and the pruning approach using note boundaries. Hopefully, we can integrate more valuable features into future Sinsy releases.

7. Acknowledgements

The authors wish to thank Dr. Shinji Sako for constructing the database. The research leading to these results was partly funded by the Strategic Information and Communications R&D Promotion Programme (SCOPE) of the Ministry of Internal Affairs and Communication, Japan.

8. References

[1] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous Modeling of Spectrum, Pitch and Du-

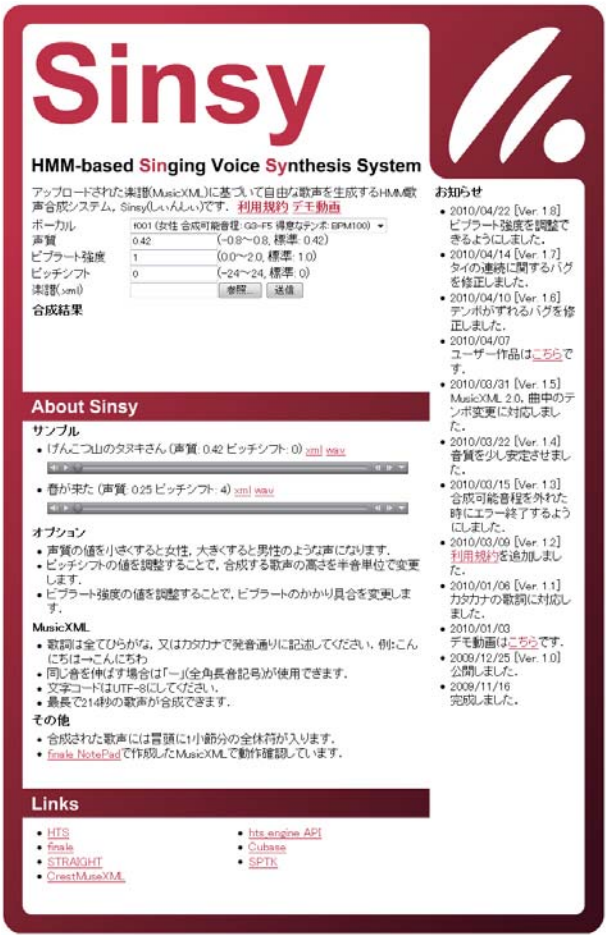


Figure 8: *HMM-based Speech Synthesis System — Sinsy.*

ration in HMM-Based Speech Synthesis," Proc. of Eurospeech, pp. 2347–2350, 1999.

[2] J. Yamagishi, "Average-Voice-Based Speech Synthesis," Ph. D. thesis, Tokyo Institute of Technology, 2006.

[3] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Speaker Interpolation in HMM-Based Speech Synthesis System," Proc. of Eurospeech, pp. 2523–2526, 1997.

[4] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Eigenvoices for HMM-Based Speech Synthesis," Proc. of ICSLP, pp. 1269–1272, 2002.

[5] K. Saino, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "An HMM-Based Singing Voice Synthesis System," Proc. of ICSLP, pp. 1141–1144, 2006.

[6] HMM-Based Singing Voice Synthesis System (Sinsy), http://www.sinsy.jp/ (in Japanese).

[7] HMM-Based Speech Synthesis System (HTS), http://hts.sp.nitech.ac.jp/.

[8] HMM-Based Speech Synthesis Engine (hts_engine API), http://hts-engine.sourceforge.net/.

[9] Speech Signal Processing Toolkit (SPTK), http://sp-tk.sourceforge.net/.

[10] A Speech Analysis, Modification and Synthesis System (STRAIGHT), http://www.wakayama-u.ac.jp/ kawahara/STRAIGHTadv/index_e.html.

[11] CrestMuseXML Toolkit (CMX), http://cmx.sourceforge.jp/.

[12] MusicXML Definition, http://musicxml.org/.

[13] K. Tokuda, T. Kobayashi, T. Chiba, and S. Imai, "Spectral Estimation of Speech by Mel-Generalized Cepstral Analysis," IEICE Trans. vol. 75-A, no. 7, pp. 1124–1134, 1992.

[14] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech Parameter Generation Algorithms for HMM-Based Speech Synthesis," Proc. of ICASSP, pp. 1315–1318, 2000.

[15] S. Imai, "Cepstral Analysis Synthesis on the Mel Frequency Scale," Proc. of ICASSP, pp. 93–96, 1983.

[16] H. Zen, T. Masuko, K. Tokuda, T. Kobayashi, and T. Kitamura, "A Hidden Semi-Markov Model-Based Speech Synthesis System," Proc. of IEICE Trans. Inf. & Sys., vol. 90D, no. 5, pp. 825–834, 2007.

[17] K. Oura, H. Zen, Y. Nankaku, A. Lee, and K. Tokuda, "A Fully Consistent Hidden Semi-Markov Model-Based Speech Recognition System," Proc. of IEICE Trans. Inf. and Syst. , vol. E91-D, no. 11, pp. 2693–2700, 208.

[18] A. Kuramatsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kawabara, and K. Shikano, "ATR Japanese Speech Database as a Tool of Speech Recognition and Synthesis," Speech Communication, vol. 9, pp. 357–363, 1990.

[19] A. Mase, K. Oura, Y. Nankaku, and K. Tokuda, "HMM-Based Singing Voice Synthesis System Using Pitch-Shifted Pseudo Training Data," Proc. of Interspeech, 2010 (to be published).

[20] K. Shinoda and T. Watanabe, "MDL-Based Context-Dependent Subword Modeling for Speech Recognition," J. Acoust. Soc. Jpn.(E), vol .21, no. 2, pp. 79–86, 2000.

[21] T. Yamada, S. Muto, Y. Nankaku, S. Sako, and K. Tokuda, "Vibrato Modeling for HMM-Based Singing Voice Synthesis," Proc. of Information Processing Society of Japan, vol. 2009-MUS-80, no. 5, pp. 1–6, 2009 (in Japanese).

[22] T. Nakano, M. Goto, and Y. Hiraga, "An Automatic Singing Skill Evaluation Method for Unknown Melodies Using Pitch Interval Accuracy and Vibrato Features", Proc. of Interspeech, pp. 1706–1709, 2006.

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

[24] C. E. Seashore, "A Musical Ornament, the Vibrato," Proc. of Psychology of Music, McGraw-Hill Book Company, pp. 33–52, 1938.

[25] S. Muto, K. Oura, Y. Nankaku, and K. Tokuda, "Reducing Computational Cost of Training for HMM-Based Singing Voice Synthesis Using Note Boundaries," Proc. of Acoustic Society of Japan Spring Meeting, vol. I, 2-7-8, pp. 347–348, 2009 (in Japanese).

[26] A New and Simplified BSD License, http://www.opensource.org/licenses/bsd-license.php.

[27] H. Kawahara, M. K. Ikuyo, and A. Cheneigne, "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," Proc. of Speech Communication, 27, pp. 187–207, 1999.

[28] H. Zen, K. Oura, T. Nose, J. Yamagishi, S. Sako, T. Toda, T. Masuko, A. W. Black, and K. Tokuda, "Recent Development of the HMM-Based Speech Synthesis System (HTS)," Proc. of APSIPA, pp. 121-130, 2009.

[29] The Hidden Markov Model Toolkit (HTK), http://htk.eng.cam.ac.uk/.

[30] T. Kitahara and H. Katayose, "On CrestMuseXML (CMX) Toolkit Ver. 0.40," IPSJ SIG Technical Report, vol. 2008-MUS-75, no. 17, pp. 95–100, 2008 (in Japanese).

[31] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi, "Hidden Markov Models Based on Multi-Space Probability Distribution for Pitch Pattern Modeling," Proc. of ICASSP, vol. I, pp. 229–232, 1999.

[32] Y. J. Wu, and R. H. Wang, "Minimum Generation Error Training for HMM-Based Speech Synthesis," Proc. of ICASSP, vol. I, pp. 89–92, 2006.

[33] T. Toda and K. Tokuda, "Speech Parameter Generation Algorithm Considering Global Variance for HMM-Based Speech Synthesis," Proc. of Interspeech, pp. 2801–2804, 2005.