AN HMM-BASED SPEECH SYNTHESIS SYSTEM APPLIED TO ENGLISH

Keiichi Tokuda[1][2]* *Heiga Zen*[1] *Alan W. Black*[2]

[1]Department of Computer Science, Nagoya Institute of Technology
[2]Language Technologies Institute, Carnegie Mellon University
{tokuda,zen}@ics.nitech.ac.jp, awb@cs.cmu.edu

ABSTRACT

This paper describes an HMM-based speech synthesis system (HTS), in which speech waveform is generated from HMMs themselves, and applies it to English speech synthesis using the general speech synthesis architecture of Festival. Similarly to other data-driven speech synthesis approaches, HTS has a compact language dependent module: a list of contextual factors. Thus, it could easily be extended to other languages, though the first version of HTS was implemented for Japanese. The resulting run-time engine of HTS has the advantage of being small: less than 1 M bytes, excluding text analysis part. Furthermore, HTS can easily change voice characteristics of synthesized speech by using a speaker adaptation technique developed for speech recognition. The relation between the HMM-based approach and other unit selection approaches is also discussed.

1. INTRODUCTION

Although many speech synthesis systems can synthesize high quality speech, they still cannot synthesize speech with various voice characteristics such as speaker individualities, speaking styles, emotions, etc. To obtain various voice characteristics in speech synthesis systems based on the selection and concatenation of acoustical units, a large amount of speech data is necessary. However, it is difficult to collect store such speech data. In order to construct speech synthesis systems which can generate various voice characteristics, the HMM-based speech synthesis system (HTS) [1] was proposed.

Figure 1 shows the system overview. In the training part, spectrum and excitation parameters are extracted from speech database and modeled by context dependent HMMs. In the synthesis part, context dependent HMMs are concatenated according to the text to be synthesized. Then spectrum and excitation parameters are generated from the HMM by using a speech parameter generation algorithm [2]. Finally, the excitation generation module and synthesis filter module synthesize speech waveform using the generated excitation and spectrum parameters. The attraction of this approach is in that voice characteristics of synthesized speech can easily be changed by transforming HMM parameters. In fact, it is shown that we can change voice characteristics of synthesized speech by applying a speaker adaptation technique [3], a speaker interpolation technique [4], or an eigenvoice technique [5].

In this paper, we applies HTS to English speech synthesis using the general speech synthesis architecture Festival [6]. Similarly to other data-driven speech synthesis approaches, HTS has a compact language dependent module: a list of contextual factors, which can be extracted through Festival (they are called “features” in Festival framework). Thus, HTS could easily be extended to other languages, though the first version of HTS was implemented for Japanese. The resulting run-time core engine of HTS has the advantage of being small: less than 1 M bytes, excluding text analysis part, and runs ten times faster than real time on a P4 machine.

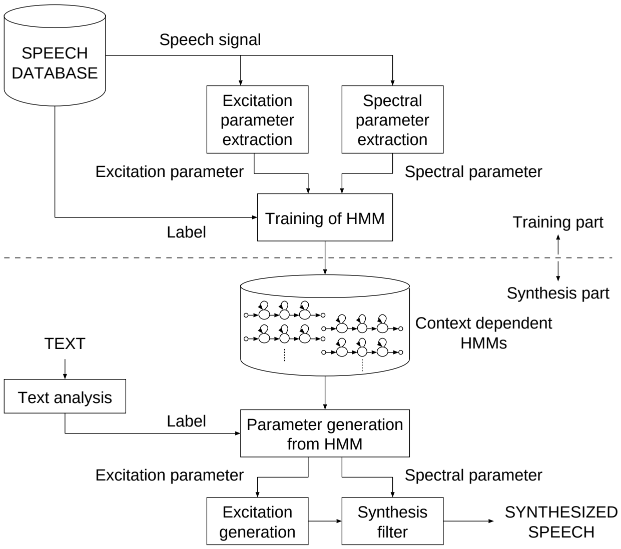

Fig. 1. HMM-based speech synthesis system.

The rest of this paper is organized as follows. Section 2 summarizes the previously proposed HMM-based speech synthesis system (HTS). Section 3 describes the language-dependent part of HTS and specifications of the resulting run-time engine, which was trained by using Festival architecture and plugged into Festival. The relation between HTS and other unit selection speech synthesis approaches is discussed in Section 4, and concluding remarks and our plans for future work are presented in the final section.

2. HMM-BASED SPEECH SYNTHESIS SYSTEM

2.1. Training Part

In HTS, output vector of HMM consists of spectrum part and excitation part. In this work, the spectrum part consists of mel-cepstral coefficient vector including the zeroth coefficients, their

[*] The first and second authors performed the work while at Carnegie Mellon University.

delta and delta-delta coefficients. On the other hand, the excitation part consists of log fundamental frequency (log F_0), its delta and delta-delta coefficients. HMMs have state duration densities to model the temporal structure of speech. As a result, HTS models not only spectrum parameter but also $F0$ and duration in a unified framework of HMM. It is noted that it does not require label boundaries for training when an appropriate initial HMM set is available because all parameters of HMMs are determined automatically through the embedded training of HMMs.

Spectrum modeling

To control the synthesis filter by HMM, its system function should be defined by the output vector of HMM, i.e., mel-cepstral coefficients. Thus we use a mel-cepstral analysis technique [7] which enables speech to be re-synthesized directly from the mel-cepstral coefficients using the MLSA (Mel Log Spectrum Approximation) filter [7][1].

F_0 modeling

The observation sequence of fundamental frequency (F_0) is composed of one-dimensional continuous values and discrete symbol which represents “unvoiced”. Therefore the conventional discrete or continuous HMMs can not be applied to F_0 pattern modeling. To model such observation sequences, we have proposed a new kind of HMM based on multi-space probability distribution (MSD-HMM) [9]. The MSD-HMM includes discrete HMM and continuous mixture HMM as special cases, and further can model the sequence of observation vectors with variable dimensionality including zero-dimensional observations, i.e., discrete symbols. As a result, MSD-HMM can model F_0 patterns without heuristic assumption.

Duration modeling

State durations of each HMM are modeled by a multivariate Gaussian distribution [10]. The dimensionality of state duration density of an HMM is equal to the number of states in the HMM, and the n-th dimension of state duration densities is corresponding to the n-th state of HMMs [2].

Decision-tree based context clustering

There are many contextual factors (e.g., phone identity factors, stress-related factors, locational factors) that affect spectrum, F_0 pattern and duration. To capture these effects, we use context-dependent HMMs. However, as contextual factors increase, their combinations also increase exponentially. Therefore, model parameters cannot be estimated accurately with limited training data. Furthermore, it is impossible to prepare speech database which includes all combinations of contextual factors. To overcome this problem, a decision-tree based context clustering technique [11, 12] is applied to distributions for spectrum, F_0 and state duration in the same manner as HMM-based speech recognition.

The decision-tree based context clustering algorithm have been extended for MSD-HMMs in [13]. Since each of spectrum, F_0 and duration has its own influential contextual factors, they are clustered independently (Fig.2). State durations of each HMM are modeled by a n-dimensional Gaussian, and context-dependent n-dimensional Gaussians are clustered by a decision tree. Note that spectrum part and F_0 part of state output vector are modeled by multivariate Gaussian distributions and multi-space probability distributions, respectively.

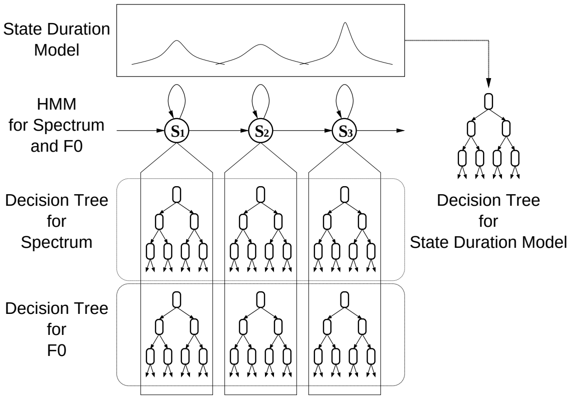

Fig. 2. Decision trees for context clustering.

Software

The training part of HTS was implemented as a modified version of HTK [14] together with SPTK [8]. Modifications which we made to HTK are listed below:

1. Context clustering based on MDL criterion (instead of ML one)
2. Stream-dependent context clustering
3. Multi-space probability distribution [9] as state output probability
4. State duration modeling

2.2. Synthesis part

In the synthesis part of HTS, first, an arbitrarily given text to be synthesized is converted to a context-based label sequence. Second, according to the label sequence, a sentence HMM is constructed by concatenating context dependent HMMs. State durations of the sentence HMM are determined so as to maximize the output probability of state durations [10], and then a sequence of mel-cepstral coefficients and $\log F_0$ values including voiced / unvoiced decisions is determined in such a way that its output probability for the HMM is maximized using the speech parameter generation algorithm (Case 1 in [2]). The main feature of the system is the use of dynamic feature: by inclusion of dynamic coefficients in the feature vector, the speech parameter sequence generated in synthesis is constrained to be realistic, as defined by the statistical parameters of the HMMs. Finally, speech waveform is synthesized directly from the generated mel-cepstral coefficients and F_0 values by using the MLSA filter. Although a mixed excitation technique for HTS was developed in [15], the traditional excitation model was used in this work.

3. HTS IMPLEMENTATION ON FESTIVAL ARCHITECTURE

We used 524 sentences from CMU Communicator database[3] for training. Speech signal was sampled at 16 kHz, windowed by a

[1] The source codes of the mel-cepstrum based vocoding technique can be found in Speech Signal Processing Toolkit (SPTK) [8].

[2] In HTS, left-to-right model with no skip is used.

[3] It can be found at `http://festvox.org/dbs/dbs_com.html`.

Table 1. Binary file size of HTS run-time engine.

module		size
decision tree	spectrum	102 kbyte
	F_0	156 kbyte
	duration	116 kbyte
distribution	spectrum	457 kbyte
	F_0	81 kbyte
	duration	39 kbyte
converter		3 kbyte
synthesizer		34 kbyte
total		988 kbyte

25-ms Blackman window with a 5-ms shift, and then mel-cepstral coefficients were obtained by the mel-cepstral analysis technique. We used 5-state left-to-right HMMs with single diagonal Gaussian output distributions. Note that each context dependent HMM corresponds to a phoneme-sized speech unit.

In this work, the following contextual factors are taken into account for English:

- phoneme:
  - {preceding, current, succeeding} phoneme
  - position of current phoneme in current syllable
- syllable:
  - number of phonemes at {preceding, current, succeeding} syllable
  - accent of {preceding, current, succeeding} syllable
  - stress of {preceding, current, succeeding} syllable
  - position of current syllable in current word
  - number of {preceding, succeeding} stressed syllables in current phrase
  - number of {preceding, succeeding} accented syllables in current phrase
  - number of syllables {from previous, to next} stressed syllable
  - number of syllables {from previous, to next} accented syllable
  - vowel within current syllable
- word:
  - guess at part of speech of {preceding, current, succeeding} word
  - number of syllables in {preceding, current, succeeding} word
  - position of current word in current phrase
  - number of {preceding, succeeding} content words in current phrase
  - number of words {from previous, to next} content word
- phrase:
  - number of syllables in {preceding, current, succeeding} phrase
  - position in major phrase
  - ToBI endtone of current phrase
- utterance:
  - number of syllables in current utterance

These factors are extracted from utterances using feature extraction functions of Festival speech synthesis system.

The whole system was trained in a few hours, and the resultant trees for spectrum models, F_0 models and state duration models had 781, 1733 and 1018 leaves in total, respectively. The run-time core engine consists of 8 modules, decision trees for spectrum, F_0 and duration, distributions of spectrum, F_0 and duration, a converter which converts features extracted by Festival into a context dependent label sequence, and a synthesizer which generates waveform for given label sequence. The binary file size of each module is shown in Table 1. Without specific efforts to compress the file size, it is already small enough for small devices such as PDAs. It was also confirmed that the core engine of HTS runs about ten times faster than real time on a P4 machine.

By listening synthesized speech samples at

`http://kt-lab.ics.nitech.ac.jp/˜zen/sound/`

it could be confirmed that the prosody is fairly natural. HTS could be used also for a prosody predictor in unit selection based speech synthesis systems.

4. DISCUSSION

Figure 3 shows the widely-used unit selection scheme [16]. In the unit selection scheme, by using the target cost and the concatenation cost, speech units are selected from the whole speech

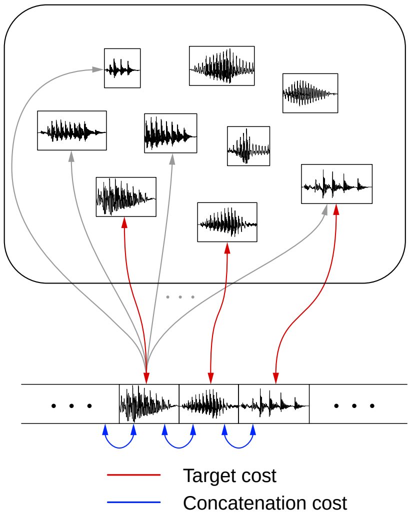

Fig. 3. Unit selection scheme.

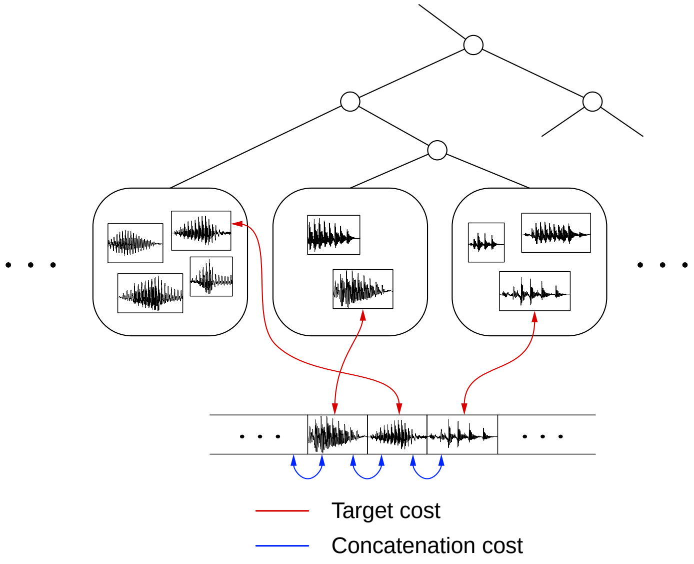

Fig. 4. Clustering-based unit selection scheme.

database, and concatenated in run-time. In this scheme, we have to define a heuristic distance between contexts to measure the target cost. To avoid this, a clustering-based scheme was also developed [17]. This approach clusters contexts in advance, and selects each unit from a cluster. To cluster speech units, some systems use the HMM-based clustering technique, e.g., [18], [19]. In this case, the structure is very similar to the HTS approach. The essential difference between the HMM-based unit selection approach and the HTS approach is in that each cluster is represented by multi-template or statistics of the cluster. It is noted that the concatenation cost corresponds to the output probability of dynamic feature parameter in the HTS approach.

Table 2 compares these two approaches. In the unit selection approach, the generated speech has a high quality at waveform level, especially in limited domain speech synthesis because it concatenates speech waveforms directly. Although unit selection approach sometimes gives excellent results, it sometimes gives very bad ones too. On the other hand, in the HTS approach, it has a quality of "vocoded speech" but sounds smooth and stable. Furthermore, it has the advantages of being small and making it possible to change voice characteristics easily by applying a speaker adaptation technique used in speech recognition. In summary, each approach has its own advantages and disadvantages.

Table 2. Relation between unit selection and generation approaches.

Unit selection	HTS
Clustering (possible use of HMM)	Clustering (use of HMM)
Multi-template	Statistics
Single tree	Multiple tree (Spectrum, F0, duration)
Advantage: • High quality at waveform level Disadvantage: • Discontinuity • Hit or miss	Disadvantage: • Vocoded speech (buzzy) Advantage: • Smooth • Stable
• Large run-time data	• Small run-time data
• Fixed voice	• Various voices

5. CONCLUSION

We have applied an HMM-based speech synthesis system (HTS) to English speech synthesis using Festival framework. The resulting run-time engine of HTS is very small: less than 1 M bytes. Furthermore, HTS can easily change voice characteristics of synthesized speech by using a speaker adaptation technique developed for speech recognition. Although synthesized speech has a typical quality of "vocoded speech," it has been shown in [15] that the mixed excitation model based on MELP speech coder [20] and postfiltering can improve the speech quality significantly.

In the near future, the suite of programs, scripts and documentation for training HMMs for HTS run-time engine and run-time plugin module for Festival will be released as a free software.

6. ACKNOWLEDGEMENTS

This work was partially supported by International Communications Foundation. Authors would like to thank members of HTS working group including Dr. Takayoshi Yoshimura for their software development efforts.

7. REFERENCES

[1] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "Simultaneous Modeling of Spectrum, Pitch and Duration in HMM-Based Speech Synthesis," Proc. of EUROSPEECH, vol.5, pp.2347–2350, 1999.

[2] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," Proc. of ICASSP 2000, vol.3, pp.1315–1318, June 2000.

[3] M. Tamura, T. Masuko, K. Tokuda, T. Kobayashi, "Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR," proc of ICASSP 2001, vol.1, pp.1–1, May 2001.

[4] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "Speaker Interpolation in HMM-Based Speech Synthesis System," Proc. of EUROSPEECH, vol.5, pp.2523–2526, 1997.

[5] K. Shichiri, A. Sawabe, K. Tokuda, T. Masuko, T. Kobayashi, T. Kitamura, "Eigenvoices for HMM-based speech synthesis," Proc. of EUROSPEECH, 2002.

[6] A. W. Black, P. Taylor and R. Caley, "The Festival Speech Sythesis System," `http://www.festvox.org/festival/`.

[7] T. Fukada, K. Tokuda, T. Kobayashi and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," Proc. of ICASSP'92, vol.1, pp.137–140, 1992.

[8] `http://kt-lab.ics.nitech.ac.jp/˜tokuda/SPTK/`.

[9] 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, 1999.

[10] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "Duration Modeling in HMM-based Speech Synthesis System," Proc. of ICSLP, vol.2, pp.29–32, 1998.

[11] J. J. Odell, "The Use of Context in Large Vocabulary Speech Recognition," PhD dissertation, Cambridge University, 1995.

[12] 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.

[13] T. Yoshimura, "Simultaneous modeling of phonetic and prosodic parameters, and characteristic conversion for HMM-based Text-To-Speech systems" PhD dissertation, Nagoya Institute of Technology, 2002.

[14] `http://htk.eng.cam.ac.uk/`.

[15] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "Mixed Excitation for HMM-based Speech Synthesis," Proc. of EUROSPEECH, vol.3, pp.2263–2266, Sep. 2001.

[16] A. W. Black and N. Campbell, "Optimising selection of units from speech databases for concatenative synthesis," Proc. EUROSPEECH, pp.581-584, Sep 1995.

[17] A. W. Black P. Taylor, "Automatically clustering similar units for unit selection in speech synthesis," Proc. EUROSPEECH, pp.601–604, Sep 1997.

[18] X. Huang, A. Acero, H. Hon, Y. Ju, J. Liu, S. Meredith and M. Plumpe, "Recent improvements on Microsoft's trainable text-to-speech system -Whistler," Proc. ICASSP, pp.959–962, 1997.

[19] R. E. Donovan and E. M. Eide, "The IBM Trainable Speech Synthesis System," Proc. of ICSLP, vol.5, pp.1703–1706, 1998.

[20] A. V. McCree, T. P. Barnwell III, "A mixed excitation LPC vocoder model for low bit rate speech coding," IEEE Trans. Speech and Audio Processing, vol.3, no.4, pp.242–250, July 1995.