Average-Voice-Based Speech Synthesis

Junichi Yamagishi

March 2006

Summary

This thesis describes a novel speech synthesis framework "Average-Voice-based Speech Synthesis." By using the speech synthesis framework, synthetic speech of arbitrary target speakers can be obtained robustly and steadily even if speech samples available for the target speaker are very small. This speech synthesis framework consists of speaker normalization algorithm for the parameter clustering, speaker normalization algorithm for the parameter estimation, the transformation/adaptation part, and modification part of the rough transformation.

In the parameter clustering using decision-tree-based context clustering techniques for average voice model, the nodes of the decision tree do not always have training data of all speakers, and some nodes have data from only one speaker. This speaker-biased node causes degradation of quality of average voice and synthetic speech after speaker adaptation, especially in prosody. Therefore, we firstly propose a new context clustering technique, named "shared-decision-tree-based context clustering" to overcome this problem. Using this technique, every node of the decision tree always has training data from all speakers included in the training speech database. As a result, we can construct decision tree common to all training speakers and each distribution of the node always reflects the statistics of all speakers.

However, when training data of each training speaker differs widely, the distributions of the node often have bias depending on speaker and/or gender and this will degrade the quality of synthetic speech. Therefore, we incorporate "speaker adaptive training" into the parameter estimation procedure of average voice model to reduce the influence of speaker dependence. In the speaker adaptive training, the speaker difference between training speaker's voice and average voice is assumed to be expressed as a simple linear regres-

sion function of mean vector of the distribution and a canonical average voice model is estimated using the assumption.

In speaker adaptation for speech synthesis, it is desirable to convert both voice characteristics and prosodic features such as F0 and phone duration. Therefore, we utilize a framework of "hidden semi-Markov model" (HSMM) which is an HMM having explicit state duration distributions and we propose an HSMM-based model adaptation algorithm to simultaneously transform both state output and state duration distributions. Furthermore, we also propose an HSMM-based speaker adaptive training algorithm to normalize both state output and state duration distributions of average voice model at the same time.

Finally, we explore several speaker adaptation algorithms to transform more effectively the average voice model into the target speaker's model when the adaptation data for the target speaker is limited. Furthermore, we adopt "MAP (Maximum A Posteriori) modification" to upgrade the estimation for the distributions having sufficient amount of speech data. When sufficient amount of the adaptation data is available, the MAP modification theoretically matches the ML estimation. As a result, it is thought that we do not need to choose the modeling strategy depending on the amount of speech data and we would accomplish the consistent method to synthesize speech in the unified way for arbitrary amount of the speech data.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Background

Since speech is obviously one of the most important ways for human to communicate, there have been a great number of efforts to incorporate speech into human-computer communication environments. As computers become more functional and prevalent, demands for technologies in speech processing area, such as speech recognition, dialogue processing, speech understanding, natural language processing, and speech synthesis, is increasing to establish high-quality human-computer communication with voice. These technologies will also be applicable to human-to-human communication with spoken language translation systems, eyes-free hands-free communication or control for handicapped persons, and so on.

Text-to-speech synthesis (TTS), one of the key technologies in speech processing, is a technique for creating speech signal from arbitrarily given text in order to transmit information from a machine to a person by voice. To fully transmit information contained in speech signals, text-to-speech synthesis systems are required to have an ability to generate natural sounding speech with arbitrary speaker's voice characteristics and various speaking styles and/or emotional expressions.

In the past decades, TTS systems based on speech unit selection and waveform concatenation techniques, such as TD-PSOLA [1], or CHATR [2], have been proposed and shown to be able to generate natural sounding

speech, and is coming widely and successfully used with the increasing availability of large speech databases.

However, it is not straightforward to make these systems have the ability of synthesizing speech with various voice characteristics, prosodic feature, speaking styles and emotional expressions. One of reasons comes from the fact that the corpus-based concatenative speech synthesis always need several large-scale speech corpora to synthesize several target speakers' voice with several speaking styles and/or emotional expressions, and consequently needs a lot of cost and takes a lot of time to record and prepare the large-scale speech data for each desired speaker of each desired speaking style. It is obvious that the realistic and desirable size of the speech data required for a new speaker and a new speaking style should be as small as we can easily obtain and prepare. One way to overcome this problem is to take advantage of a large amount of speech corpora of speakers which have been already recorded and prepared and to apply voice or speaking style conversion techniques to a target speaker's speech or a target speaking style. Using the voice conversion techniques (e.g. [3]), we can manage the problem. However, most of the techniques does not focus on the precise conversion of the prosodic features such as fundamental frequency and phone duration, although the fundamental frequency and duration features as well as spectral features affect speaker characteristics [4], [5]. Moreover, it does lack consistency of the policy. In other words, we have to choose the strategy depending on the amount of available speech data.

On the other hand, an HMM-based speech synthesis system with speaker adaptation and "average voice model" [6]–[8] can simultaneously transform voice characteristics and fundamental frequency of synthetic speech into those of a target speaker by using a small amount of speech data uttered by the target speaker, and becomes a promising approach to overcoming this problem. However, the quality of synthetic speech using the method is hardly adequate compared to a speaker-dependent HMM-based speech synthesis system [9], [10].

In that system, the initial model of the transformation, namely the average voice model, crucially affects quality of synthetic speech generated from adapted models. To obtain higher performance to a wide variety of target

speakers, the average voice model should not have any bias depending on speaker and/or gender. However, it would occur that the distributions and structure of model topology of the average voice model have relatively large bias depending on speaker and/or gender included in the training speech database. This will affect model adaptation performance and degrade the quality of synthetic speech. Therefore, we need to consider effective normalization techniques of speaker characteristics of the training speakers in the training stage of the average voice model.

1.2 Scope of Thesis

In this thesis, we describe a novel speech synthesis framework"Average-Voice-based Speech Synthesis." By using the speech synthesis framework, synthetic speech of the target speaker can be obtained robustly and steadily even if speech samples available for the target speaker are very small. This speech synthesis framework consists of speaker normalization algorithm for the parameter clustering, speaker normalization algorithm for the parameter estimation, the transformation/adaptation part, and modification part of the rough transformation.

In the decision-tree-based context clustering for average voice model, the nodes of the decision tree do not always have training data of all speakers, and some nodes have data from only one speaker. This speaker-biased node causes degradation of quality of average voice and synthetic speech after speaker adaptation, especially in prosody. In chapter 4, we propose a new context clustering technique, named "shared-decision-tree-based context clustering" to overcome this problem. Using this technique, every node of the decision tree always has training data from all speakers included in the training speech database. As a result, we can construct decision tree common to all training speakers and each distribution of the node always reflects the statistics of all speakers.

However, when training data of each training speaker differs widely, the distributions of the node often have bias depending on speaker and/or gender and this will degrade the quality of synthetic speech. Therefore, in chapter 5, we incorporate "speaker adaptive training" into the parameter estimation

procedure of average voice model to reduce the influence of speaker dependence. In the speaker adaptive training, the speaker difference between training speaker's voice and average voice is assumed to be expressed as a simple linear regression function of mean vector of the distribution and a canonical average voice model is estimated using the assumption.

In speaker adaptation for speech synthesis, it is desirable to convert both voice characteristics and prosodic features such as F0 and phone duration. Therefore, in chapter 6, we utilize a framework of "hidden semi-Markov model" (HSMM) which is an HMM having explicit state duration distributions and we propose an HSMM-based model adaptation algorithm to simultaneously transform both state output and state duration distributions. Furthermore, we also propose an HSMM-based speaker adaptive training algorithm to normalize both state output and state duration distributions of average voice model at the same time.

In chapter 7, we explore several speaker adaptation algorithms to transform more effectively the average voice model into the target speaker's model when the adaptation data for the target speaker is limited. Furthermore, we adopt "MAP (Maximum A Posteriori) modification" to upgrade the estimation for the distributions having sufficient amount of speech data. When sufficient amount of the adaptation data is available, the MAP modification theoretically matches the ML estimation. As a result, it is thought that we do not need to choose the modeling strategy depending on the amount of speech data and we would accomplish the consistent method to synthesize speech in the unified way for arbitrary amount of the speech data.

Chapter 2

The Hidden Markov Model

The hidden Markov model (HMM) [11]–[13] is one of statistical time series models widely used in various fields. Especially, speech recognition systems to recognize time series sequences of speech parameters as digit, character, word, or sentence can achieve success by using several refined algorithms of the HMM. Furthermore, text-to-speech synthesis systems to generate speech from input text information has also made substantial progress by using the excellent framework of the HMM. In this chapter, we briefly describe the basic theory of the HMM.

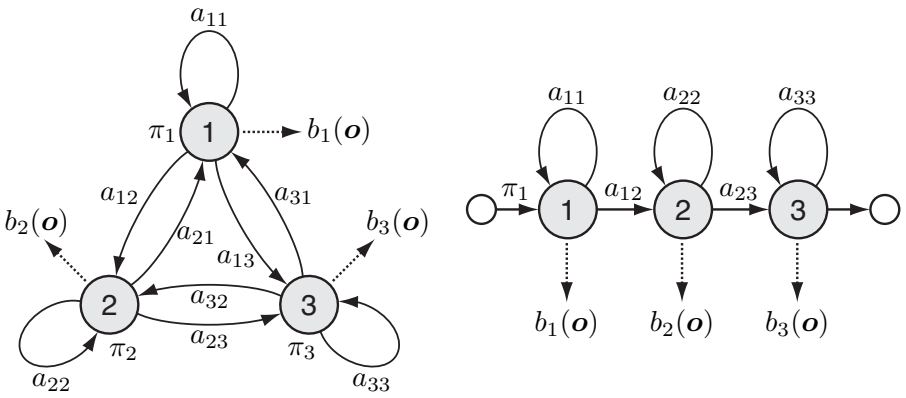
2.1 Definition

A hidden Markov model (HMM) is a finite state machine which generates a sequence of discrete time observations. At each time unit, the HMM changes states at Markov process in accordance with a state transition probability, and then generates observational data $\boldsymbol{o}$ in accordance with an output probability distribution of the current state.

An N-state HMM is defined by the state transition probability $\boldsymbol{A} = \{a_{ij}\}_{i,j=1}^{N}$, the output probability distribution $\boldsymbol{B} = \{b_i(\boldsymbol{o})\}_{i=1}^{N}$, and initial state probability $\boldsymbol{\Pi} = \{\pi_i\}_{i=1}^{N}$. For notational simplicity, we denote the model parameters of the HMM as follow:

$$\lambda = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}). \tag{2.1}$$

Figure 2.1 shows examples of typical HMM structure. Figure 2.1 (a)



(a) A 3-state ergodic model (b) A 3-state left-to-right model

Figure 2.1: Examples of HMM structure.

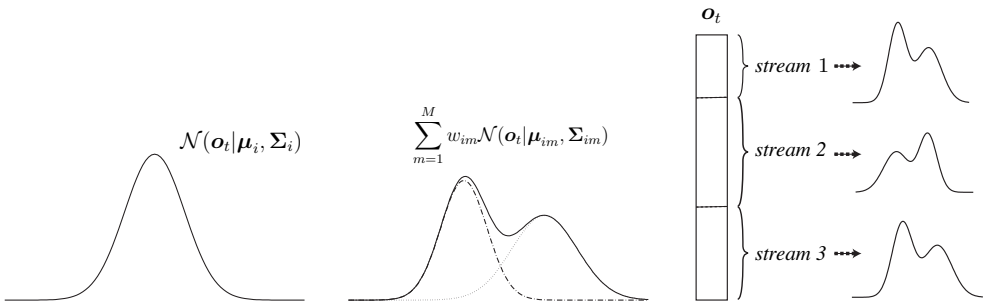
shows a 3-state ergodic model, in which each state of the model can be reached from every other state of the model in a single transition, and Fig. 2.1 (b) shows a 3-state left-to-right model, in which the state index simply increases or stays depending on time increment. The left-to-right models are often used as speech units to model speech parameter sequences since they can appropriately model signals whose properties successively change.

The output probability distribution $b_i(\boldsymbol{o})$ of the observational data $\boldsymbol{o}$ of state i can be discrete or continuous depending on the observations. In continuous distribution HMM (CD-HMM) for the continuous observational data, the output probability distribution is usually modeled by a mixture of multivariate Gaussian distributions as follows:

$$b_i(\boldsymbol{o}) = \sum_{m=1}^{M} w_{im}\, \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) \tag{2.2}$$

where M is the number of mixture components for the distribution, and w_{im}, $\boldsymbol{\mu}_{im}$ and $\boldsymbol{\Sigma}_{im}$ are a weight, a L-dimensional mean vector, and a $L \times L$ covariance matrix of mixture component m of state i, respectively. A Gaussian distribution $\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im})$ of each component is defined by

$$\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_{im}, \boldsymbol{\Sigma}_{im}) = \frac{1}{\sqrt{(2\pi)^L |\boldsymbol{\Sigma}_{im}|}} \exp\left(-\frac{1}{2}(\boldsymbol{o} - \boldsymbol{\mu}_{im})^\top \boldsymbol{\Sigma}_{im}^{-1}(\boldsymbol{o} - \boldsymbol{\mu}_{im})\right), \tag{2.3}$$



(a) Gaussian PDF (b) Multi Mixture PDF (c) Multi Stream PDF

Figure 2.2: Output distributions.

where L is the dimensionality of the observation data $\boldsymbol{o}$. Mixture weights w_{im} satisfy the following stochastic constraint

$$\sum_{m=1}^{M} w_{im} = 1, \qquad 1 \leq i \leq N \tag{2.4}$$

$$w_{im} \geq 0, \qquad 1 \leq i \leq N, \quad 1 \leq m \leq M \tag{2.5}$$

so that $b_i(\boldsymbol{o})$ are properly normalized as probability density function, i.e.,

$$\int_{\boldsymbol{o}} b_i(\boldsymbol{o}) d\boldsymbol{o} = 1, \qquad 1 \leq i \leq N. \tag{2.6}$$

When the observation vector $\boldsymbol{o}_t$ is divided into S stochastic-independent data streams, i.e., $\boldsymbol{o} = [\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_S^\top]^\top$, $b_i(\boldsymbol{o})$ is formulated by product of Gaussian mixture densities,

$$b_i(\boldsymbol{o}) = \prod_{s=1}^{S} b_{is}(\boldsymbol{o}_s) \tag{2.7}$$

$$= \prod_{s=1}^{S} \left\{ \sum_{m=1}^{M_s} w_{ism} \mathcal{N}(\boldsymbol{o}_s; \boldsymbol{\mu}_{ism}, \boldsymbol{\Sigma}_{ism}) \right\} \tag{2.8}$$

where M_s is the number of components in stream s, and w_{ism}, $\boldsymbol{\mu}_{ism}$ and $\boldsymbol{\Sigma}_{ism}$ are a weight, a L-dimensional mean vector, and a $L \times L$ covariance matrix of mixture component m of state i in stream s, respectively (Fig. 2.2).

2.1.1 Probability Evaluation

When a state sequence of length T is determined as $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$, the observation probability of an observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T)$ of

length T, given the HMM λ can be simply calculated by multiplying the output probabilities for each state, that is,

$$P(\boldsymbol{O}|\boldsymbol{q},\lambda) = \prod_{t=1}^{T} P(\boldsymbol{o}_t|q_t,\lambda) = \prod_{t=1}^{T} b_{q_t}(\boldsymbol{o}_t). \tag{2.9}$$

The probability of such a a state sequence $\boldsymbol{q}$ can be calculated by multiplying the state transition probabilities,

$$P(\boldsymbol{q}|\lambda) = \prod_{t=1}^{T} a_{q_{t-1}q_t} \tag{2.10}$$

where $a_{q_0 i} = \pi_i$ is the initial state probability. Using Bayes' theorem, the joint probability of $\boldsymbol{O}$ and $\boldsymbol{q}$ can be simply written as

$$P(\boldsymbol{O},\boldsymbol{q}|\lambda) = P(\boldsymbol{O}|\boldsymbol{q},\lambda)P(\boldsymbol{q}|\lambda). \tag{2.11}$$

Hence, the probability of the observation sequence $\boldsymbol{O}$ given the HMM λ is calculated by using marginalization of state sequences $\boldsymbol{q}$, that is, by summing $P(\boldsymbol{O},\boldsymbol{q}|\lambda)$ over all possible state sequences $\boldsymbol{q}$,

$$P(\boldsymbol{O}|\lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O},\boldsymbol{q}|\lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}|\boldsymbol{q},\lambda)P(\boldsymbol{q}|\lambda) \tag{2.12}$$

$$= \sum_{\text{all } \boldsymbol{q}} \prod_{t=1}^{T} a_{q_{t-1}} a_{q_t} b_{q_t}(\boldsymbol{o}_t). \tag{2.13}$$

Considering that the state sequences become trellis structure, this probability of the observation sequence can be transformed as follows:

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} P(\boldsymbol{o}_1,\ldots,\boldsymbol{o}_t, q_t = i \,|\, \lambda) \cdot P(\boldsymbol{o}_{t+1},\ldots,\boldsymbol{o}_T \,|\, q_t = i,\lambda) \tag{2.14}$$

for $\forall\ t \in [1,T]$. Therefore, we can efficiently calculate the probability of the observation sequence (Eq. 2.13) using forward and backward probabilities defined as

$$\alpha_t(i) = P(\boldsymbol{o}_1,\boldsymbol{o}_2,\ldots,\boldsymbol{o}_t, q_t = i \,|\, \lambda), \tag{2.15}$$

$$\beta_t(i) = P(\boldsymbol{o}_{t+1},\boldsymbol{o}_{t+2},\ldots,\boldsymbol{o}_T \,|\, q_t = i,\lambda). \tag{2.16}$$

The forward and/or backward probabilities can be recursively calculated as follows:

1. Initialization

$$\alpha_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \le i \le N \tag{2.17}$$

$$\beta_T(i) = 1 \qquad 1 \le i \le N. \tag{2.18}$$

2. Recursion

$$\alpha_{t+1}(i) = \left[\sum_{j=1}^{N} \alpha_t(j) a_{ji}\right] b_i(\boldsymbol{o}_{t+1}), \qquad \begin{array}{l} 1 \le i \le N, \\ t = 2, \ldots, T \end{array} \tag{2.19}$$

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\boldsymbol{o}_{t+1}) \beta_{t+1}(j), \qquad \begin{array}{l} 1 \le i \le N, \\ t = T-1, \ldots, 1. \end{array} \tag{2.20}$$

Thus, the $P(\boldsymbol{O}|\lambda)$ is given by

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \tag{2.21}$$

for $\forall\ t \in [1, T]$.

2.2 Optimal State Sequence

A single best state sequence $\boldsymbol{q}^* = (q_1^*, q_2^*, \ldots, q_T^*)$ for a given observation sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, \ldots, \boldsymbol{o}_T)$ is also useful for various applications. For instance, most speech recognition systems use the joint probability of the observation sequence and the most likely state sequence $P(\boldsymbol{O}, \boldsymbol{q}^*|\lambda)$ to approximate the real probability $P(\boldsymbol{O}|\lambda)$

$$P(\boldsymbol{O}|\lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda) \tag{2.22}$$

$$\simeq \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda). \tag{2.23}$$

The best state sequence $\boldsymbol{q}^* = \text{argmax}_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda)$ can be obtained by a manner similar to the Dynamic Programming (DP) procedure, which is often referred to as the Viterbi algorithm. Let $\delta_t(i)$ be the probability of the most likely state sequence ending in state i at time t

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P(\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t, q_1, \ldots, q_{t-1}, q_t = i|\lambda), \tag{2.24}$$

and $\psi_t(i)$ be the array to keep track. Using these variables, the Viterbi algorithm can be written as follows:

1. Initialization

$$\delta_1(i) = \pi_i b_i(\boldsymbol{o}_1), \qquad 1 \leq i \leq N, \tag{2.25}$$

$$\psi_1(i) = 0, \qquad 1 \leq i \leq N. \tag{2.26}$$

2. Recursion

$$\delta_t(j) = \max_i \left[\delta_t(i) a_{ij}\right] \boldsymbol{o}_t, \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T \end{array} \tag{2.27}$$

$$\psi_t(j) = \operatorname*{argmax}_i \left[\delta_t(i) a_{ij}\right], \qquad \begin{array}{l} 1 \leq i \leq N, \\ t = 2, \ldots, T. \end{array} \tag{2.28}$$

3. Termination

$$P(\boldsymbol{O}, \boldsymbol{q}^* | \lambda) = \max_i \left[\delta_T(i)\right], \tag{2.29}$$

$$q_T^* = \operatorname*{argmax}_i [\delta_T(i)]. \tag{2.30}$$

4. Path backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*). \tag{2.31}$$

2.3 Parameter Estimation

There is no known way to analytically solve the model parameter set which satisfies a certain optimization criterion such as maximum likelihood (ML) criterion as follows:

$$\lambda^* = \operatorname*{argmax}_\lambda P(\boldsymbol{O}|\lambda) \tag{2.32}$$

$$= \operatorname*{argmax}_\lambda \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda). \tag{2.33}$$

Since this problem is an optimization problem from incomplete data including the hidden variable $\boldsymbol{q}$, it is difficult to determine λ^* which globally maximizes likelihood $P(\boldsymbol{O}|\lambda)$ for a given observation sequence $\boldsymbol{O}$ in a closed form.

However, a model parameter set λ which locally maximizes $P(\boldsymbol{O}|\lambda)$ can be obtained using an iterative procedure such as the expectation-maximization (EM) algorithm which conducts optimization of the complete dataset. This optimization algorithm is often referred to as the Baum-Welch algorithm.

In the following, the EM algorithm for the CD-HMM using a single Gaussian distribution are described. The EM algorithm for the HMM with discrete output distributions or Gaussian mixture distributions can also be derived straightforwardly.

2.3.1 Auxiliary Function Q

In the EM algorithm, an auxiliary function $Q(\lambda', \lambda)$ of current parameter set λ' and new parameter set λ is defined as follows:

$$Q(\lambda', \lambda) = \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{q}|\boldsymbol{O}, \lambda') \log P(\boldsymbol{O}, \boldsymbol{q}|\lambda). \tag{2.34}$$

At each iteration of the procedure, current parameter set λ' is replaced by new parameter set λ which maximizes $Q(\lambda', \lambda)$. This iterative procedure can be proved to increase likelihood $P(\boldsymbol{O}|\lambda)$ monotonically and converge to a certain critical point, since it can be proved that the Q-function satisfies the following theorems:

- Theorem 1

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(\boldsymbol{O}|\lambda) \geq P(\boldsymbol{O}|\lambda') \tag{2.35}$$

- Theorem 2
  The auxiliary function $Q(\lambda', \lambda)$ has an unique global maximum as a function of λ, and this maximum is the one and only critical point.

- Theorem 3
  A parameter set λ is a critical point of the likelihood $P(\boldsymbol{O}|\lambda)$ if and only if it is a critical point of the Q-function.

2.3.2 Maximization of Q-Function

Using Eq. (2.13), logarithm of likelihood function of $P(\boldsymbol{O}, \boldsymbol{q}|\lambda)$ can be written as

$$\log P(\boldsymbol{O}, \boldsymbol{q}|\lambda) = \sum_{t=1}^{T} \log a_{q_{t-1}q_t} + \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}), \tag{2.36}$$

where $a_{q_0 q_1}$ denotes π_{q_1}. The Q-function (Eq. (2.34)) can be written as

$$Q(\lambda', \lambda) = \sum_{i=1}^{N} P(\boldsymbol{O}, q_1 = i|\lambda') \log \pi_i \tag{2.37}$$

$$+ \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{t=1}^{T-1} P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda') \log a_{ij} \tag{2.38}$$

$$+ \sum_{i=1}^{N} \sum_{t=1}^{T} P(\boldsymbol{O}, q_t = i|\lambda) \log \mathcal{N}(\boldsymbol{o}_t; \boldsymbol{\mu}_{q_t}, \boldsymbol{\Sigma}_{q_t}). \tag{2.39}$$

The parameter set λ which maximizes above the Q-function subject to the stochastic constraints $\sum_{i=1}^{N} \pi_i = 1$ and $\sum_{j=1}^{N} a_{ij} = 1$ for $1 \leq i \leq N$ can be derived by using Lagrange multipliers method of Eqs. (2.37)–(2.38) and partial differential equation of Eq. (2.39):

$$\pi_i = \gamma_1(i), \tag{2.40}$$

$$a_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \tag{2.41}$$

$$\boldsymbol{\mu}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot \boldsymbol{o}_t}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{2.42}$$

$$\boldsymbol{\Sigma}_i = \frac{\sum_{t=1}^{T} \gamma_t(i) \cdot (\boldsymbol{o}_t - \boldsymbol{\mu}_i)(\boldsymbol{o}_t - \boldsymbol{\mu}_i)^\top}{\sum_{t=1}^{T} \gamma_t(i)}, \tag{2.43}$$

where $\gamma_t(i)$ and $\xi_t(i,j)$ are the state occupancy probability of being state i at time t, and the probability of being state i at time t and state j at time $t+1$, respectively,

$$\gamma_t(i) = P(\boldsymbol{O}, q_t = i|\lambda) \tag{2.44}$$

$$= \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}, \tag{2.45}$$

$$\xi_t(i,j) = P(\boldsymbol{O}, q_t = i, q_{t+1} = j|\lambda) \tag{2.46}$$

$$= \frac{\alpha_t(i)a_{ij}b_j(\boldsymbol{o}_{t+1})\beta_{t+1}(j)}{\sum_{l=1}^{N}\sum_{n=1}^{N} \alpha_t(l)a_{ln}b_n(\boldsymbol{o}_{t+1})\beta_{t+1}(n)}. \tag{2.47}$$

Chapter 3

HMM-Based Speech Synthesis

This chapter describes an HMM-based text-to-speech synthesis (TTS) system [14] [10]. In the HMM-based speech synthesis, the speech parameters of a speech unit such as the spectrum, fundamental frequency (F0), and phoneme duration are statistically modeled and generated by using HMMs based on maximum likelihood criterion [9], [15]–[17]. In this chapter, we briefly describe the basic structure and the algorithms of the HMM-based TTS system.

3.1 Parameter Generation Algorithm

3.1.1 Formulation of the Problem

First, we describe an algorithm to directly generate optimal speech parameters from the HMM in the maximum likelihood sense [9], [15], [16]. Given a HMM λ using continuous distributions and length T of a parameter sequence to be generated, the problem for generating the speech parameters from the HMM is to obtain a speech parameter vector sequence $\boldsymbol{O} = (\boldsymbol{o}_1, \boldsymbol{o}_2, , \ldots, \boldsymbol{o}_T)$ which maximizes $P(\boldsymbol{O}|\lambda, T)$ with respect to $\boldsymbol{O}$,

$$\boldsymbol{O}^* = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T) \tag{3.1}$$

$$= \underset{\boldsymbol{O}}{\operatorname{argmax}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T). \tag{3.2}$$

Since there is no known method to analytically obtain the speech parameter sequence which maximizes $P(\boldsymbol{O}|\lambda, T)$ in a closed form, this problem is approximated[1] by using the most likely state sequence in the same manner as the Viterbi algorithm, i.e.,

$$\boldsymbol{O}^* = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, T) \tag{3.3}$$

$$= \underset{\boldsymbol{O}}{\operatorname{argmax}} \sum_{\text{all } \boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T) \tag{3.4}$$

$$\simeq \underset{\boldsymbol{O}}{\operatorname{argmax}} \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T). \tag{3.5}$$

Using Bayes' theorem, the joint probability of $\boldsymbol{O}$ and $\boldsymbol{q}$ can be simply written as

$$\boldsymbol{O}^* \simeq \underset{\boldsymbol{O}}{\operatorname{argmax}} \max_{\boldsymbol{q}} P(\boldsymbol{O}, \boldsymbol{q}|\lambda, T) \tag{3.6}$$

$$= \underset{\boldsymbol{O}}{\operatorname{argmax}} \max_{\boldsymbol{q}} P(\boldsymbol{O}|\boldsymbol{q}, \lambda, T) P(\boldsymbol{q}|\lambda, T). \tag{3.7}$$

Hence, the optimization problem of the probability of the observation sequence $\boldsymbol{O}$ given the HMM λ and the length T is divided into the following two optimization problems:

$$\boldsymbol{q}^* = \underset{\boldsymbol{q}}{\operatorname{argmax}}\, P(\boldsymbol{q}|\lambda, T) \tag{3.8}$$

$$\boldsymbol{O}^* = \underset{\boldsymbol{O}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T). \tag{3.9}$$

If the parameter vector at frame t is determined independently of preceding and succeeding frames, the speech parameter sequence $\boldsymbol{O}$ which maximizes $P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)$ is obtained as a sequence of mean vectors of the given optimum state sequence $\boldsymbol{q}^*$. This will cause discontinuity in the generated spectral sequence at transitions of states, resulting in clicks in synthesized speech which degrade quality of synthesized speech. To avoid this, it is assumed that the speech parameter vector $\boldsymbol{o}_t$ consists of the M-dimensional static feature vector $\boldsymbol{c}_t = [c_t(1), c_t(2), \ldots, c_t(M)]^\top$ (e.g., cepstral coefficients) and the M-dimensional dynamic feature vectors $\Delta\boldsymbol{c}_t$, $\Delta^2\boldsymbol{c}_t$ (e.g., delta and delta-delta cepstral coefficients), i.e., $\boldsymbol{o}_t = \left[\boldsymbol{c}_t^\top, \Delta\boldsymbol{c}_t^\top, \Delta^2\boldsymbol{c}_t^\top\right]^\top$ and that the dynamic

[1] An algorithm to obtain $\boldsymbol{O}$ which maximizes $P(\boldsymbol{O}|\lambda)$ using EM algorithm is shown in [18].

feature vectors are determined by linear combination of the static feature vectors of several frames around the current frame. By setting $\Delta^{(0)}\boldsymbol{c}_t = \boldsymbol{c}_t$, $\Delta^{(1)}\boldsymbol{c}_t = \Delta\boldsymbol{c}_t$, and $\Delta^{(2)}\boldsymbol{c}_t = \Delta^2\boldsymbol{c}_t$, the general form $\Delta^{(n)}\boldsymbol{c}_t$ is defined as

$$\Delta^{(n)}\boldsymbol{c}_t = \sum_{\tau=-L_-^{(n)}}^{L_+^{(n)}} w_{t+\tau}^{(n)}\boldsymbol{c}_t \quad 0 \le n \le 2, \tag{3.10}$$

where $L_-^{(0)} = L_+^{(0)} = 0$ and $w_0^{(0)} = 1$. Then, the optimization problem of the observation sequence $\boldsymbol{O}$ is considered to be maximizing $P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)$ with respect to $\boldsymbol{C} = (\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_T)$ under the constraints Eq. (3.10).

3.1.2 Solution for the Optimization Problem $\boldsymbol{O}^*$

First, we describe a solution for the optimization problem $\boldsymbol{O}^*$ given the optimum state sequence $\boldsymbol{q}^*$. The speech parameter vector sequence $\boldsymbol{O}$ is rewritten in a vector form as $\boldsymbol{O} = \left[\boldsymbol{o}_1^\top, \boldsymbol{o}_2^\top, \ldots, \boldsymbol{o}_T^\top\right]^\top$, that is, $\boldsymbol{O}$ is a super-vector made from all of the parameter vectors. In the same way, $\boldsymbol{C}$ is rewritten as $\boldsymbol{C} = \left[\boldsymbol{c}_1^\top, \boldsymbol{c}_2,^\top, \ldots, \boldsymbol{c}_T^\top\right]^\top$. Then, $\boldsymbol{O}$ can be expressed by $\boldsymbol{C}$ as $\boldsymbol{O} = \boldsymbol{W}\boldsymbol{C}$ where

$$\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_T]^\top \tag{3.11}$$

$$\boldsymbol{w}_t = [\boldsymbol{w}_t^{(0)}, \boldsymbol{w}_t^{(1)}, \boldsymbol{w}_t^{(2)}] \tag{3.12}$$

$$\begin{aligned}\boldsymbol{w}_t^{(n)} = [&\underbrace{\boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}}_{1\text{st}}, \\ &\underbrace{w^{(n)}(-L_-^{(n)})\boldsymbol{I}_{M\times M}, \ldots, w^{(n)}(0)\boldsymbol{I}_{M\times M}, \ldots, w^{(n)}(L_+^{(n)})\boldsymbol{I}_{M\times M}}_{(t-L_-^{(n)})\text{-th} \qquad t\text{-th} \qquad (t+L_+^{(n)})\text{-th}}, \\ &\underbrace{\boldsymbol{0}_{M\times M}, \ldots, \boldsymbol{0}_{M\times M}}_{T\text{-th}}]^\top, \qquad n = 0, 1, 2,\end{aligned} \tag{3.13}$$

and $\boldsymbol{0}_{M\times M}$ and $\boldsymbol{I}_{M\times M}$ are the $M \times M$ zero matrix and the $M \times M$ identity matrix, respectively. It is assumed that $\boldsymbol{c}_t = \boldsymbol{0}_M$ $(t < 1, T < t)$ where $\boldsymbol{0}_M$ denotes the M-dimensional zero vector. Using the variable, the probability $P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)$ is written as

$$P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T) = P(\boldsymbol{W}\boldsymbol{C}|\boldsymbol{q}^*, \lambda, T) \tag{3.14}$$

$$= \frac{1}{\sqrt{(2\pi)^{3MT}|\boldsymbol{\Sigma}|}} \exp\left(-\frac{1}{2}(\boldsymbol{W}\boldsymbol{C} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{W}\boldsymbol{C} - \boldsymbol{\mu})\right), \tag{3.15}$$

where $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_{q_1^*}^\top, \boldsymbol{\mu}_{q_2^*}^\top, \ldots, \boldsymbol{\mu}_{q_T^*}^\top\right]^\top$ and $\boldsymbol{U} = \mathrm{diag}[\boldsymbol{U}_{q_1^*}, \boldsymbol{U}_{q_2^*}, \ldots, \boldsymbol{U}_{q_T^*}]$, and $\boldsymbol{\mu}_{q_t^*}$ and $\boldsymbol{U}_{q_t^*}$ are the mean vector and the diagonal covariance matrix of the state q_t of the optimum state sequence $\boldsymbol{q}^*$. Thus, by setting

$$\frac{\partial P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)}{\partial \boldsymbol{C}} = \boldsymbol{0}_{TM\times 1}, \tag{3.16}$$

the following equations are obtained,

$$\boldsymbol{R}\boldsymbol{C} = \boldsymbol{r}, \tag{3.17}$$

where $TM \times TM$ matrix $\boldsymbol{R}$ and TM-dimensional vector $\boldsymbol{r}$ are as follows:

$$\boldsymbol{R} = \boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{W}, \tag{3.18}$$

$$\boldsymbol{r} = \boldsymbol{W}^\top \boldsymbol{U}^{-1} \boldsymbol{\mu}. \tag{3.19}$$

By solving Eq. (3.17), a speech parameter sequence $\boldsymbol{C}$ which maximizes $P(\boldsymbol{O}|\boldsymbol{q}^*, \lambda, T)$ is obtained. By utilizing the special structure of $\boldsymbol{R}$, Eq. (3.17) can be solved by the Cholesky decomposition or the QR decomposition efficiently.

3.1.3 Solution for the Optimization Problem $\boldsymbol{q}^*$

Next, we describe a solution for the optimization problem $\boldsymbol{q}^*$ given the model parameter λ and the length T. The $P(\boldsymbol{q}|\lambda, T)$ is calculated as

$$P(\boldsymbol{q}|\lambda, T) = \prod_{t=1}^{T} a_{q_{t-1}q_t} \tag{3.20}$$

where $a_{q_0 q_1} = \pi_{q_1}$. If the value of $P(\boldsymbol{q}|\lambda, T)$ for every possible sequence $\boldsymbol{q}$ can be obtained, we can solve the optimization problem. However, it is impractical because there are too many combinations of $\boldsymbol{q}$. Furthermore, if state duration is controlled only by self-transition probability, state duration probability density associated with state i becomes the following geometrical distribution:

$$p_i(d) = (a_{ii})^{d-1}(1 - a_{ii}), \tag{3.21}$$

where $p_i(d)$ represents probability of d consecutive observations in state i, and a_{ii} is self-transition probability associated with sate i. This exponential state duration probability density is inappropriate for controlling state

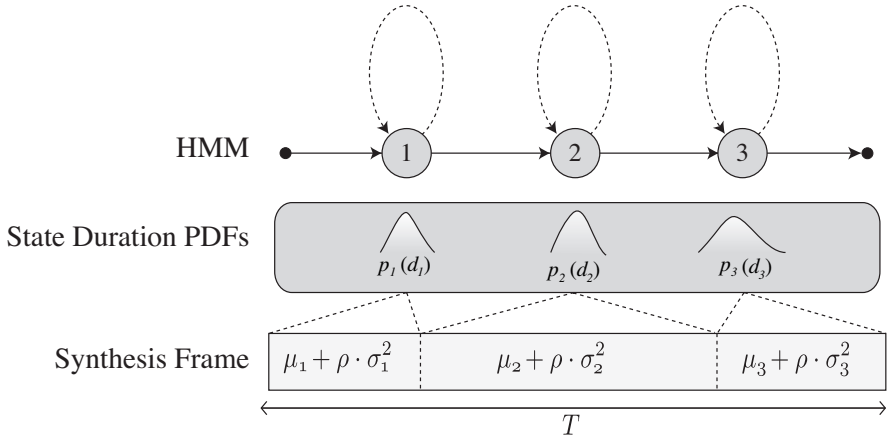


Figure 3.1: Duration synthesis

and/or phoneme duration. To control temporal structure appropriately, HMMs should have explicit state duration distributions. The state duration distributions can be modeled by parametric probability density functions (pdfs) such as the Gaussian pdfs or Gamma pdfs or Poisson pdfs.

Assume that the HMM λ is left-to-right model with no skip, then the probability of the state sequence $\boldsymbol{q} = (q_1, q_2, \ldots, q_T)$ is characterized only by explicit state duration distributions. Let $p_k(d_k)$ be the probability of being d_k frames at state k, then the probability of the state sequence $\boldsymbol{q}$ can be written as

$$P(\boldsymbol{q}|\lambda, T) = \prod_{k=1}^{K} p_k(d_k) \tag{3.22}$$

where K is the total number of states visited during T frames, and

$$\sum_{k=1}^{K} d_{q_k} = T. \tag{3.23}$$

When the state duration probability density is modeled by a single Gaussian pdf,

$$p_k(d_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{(d_k - m_k)^2}{2\sigma_k^2}\right), \tag{3.24}$$

$\boldsymbol{q}^*$ which maximizes $P(\boldsymbol{q}|\lambda, T)$ under the constraint Eq. (3.23) is obtained by

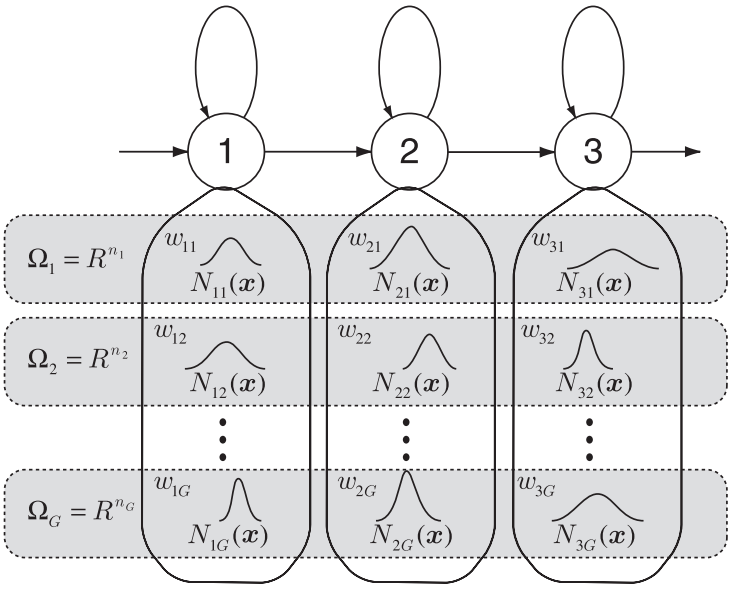


Figure 3.2: MSD-HMM

using Lagrange multipliers method of Eq. (3.22):

$$d_k = m_k + \rho \cdot \sigma_k^2, \qquad 1 \le k \le K, \tag{3.25}$$

$$\rho = \left(T - \sum_{k=1}^{K} m_k \right) \Bigg/ \sum_{k=1}^{K} \sigma_k^2, \tag{3.26}$$

where m_k and σ_k are the mean and variance of the duration distribution of state k, respectively (Fig. 3.1).

3.2 Multi-Space Probability Distribution

In order to synthesize speech, it is necessary to model and generate fundamental frequency (F0) patterns as well as spectral sequences. However, the F0 patterns cannot be modeled by conventional discrete or continuous HMMs, because the values of F0 are not defined in unvoiced regions, i.e., the observation sequence of an F0 pattern is composed of one-dimensional continuous values and a discrete symbol which represents "unvoiced." Therefore we apply multi-space probability distribution HMM (MSD-HMM) [19]–[21] to F0 pattern modeling and generation.

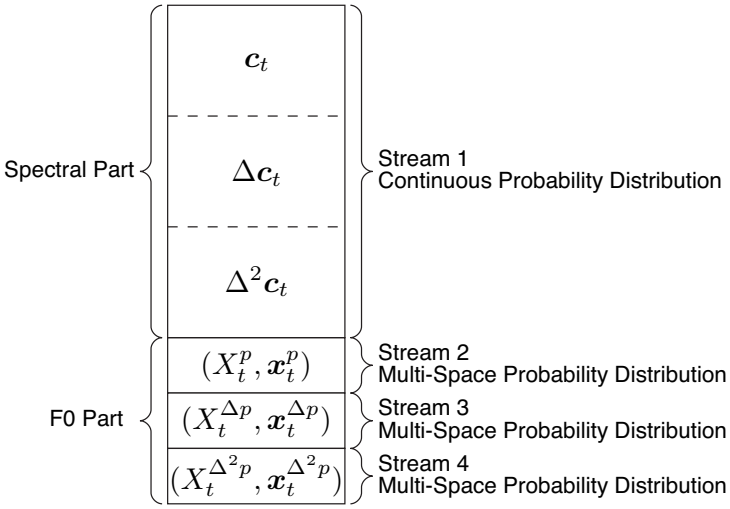


Figure 3.3: Observation vector

In the MSD-HMM, the observation sequence of F0 pattern is viewed as a mixed sequence of outputs from a one-dimensional space Ω_1 and a zero-dimensional space Ω_2 which correspond to voiced and unvoiced regions, respectively. Each space has the space weight w_g ($\sum_{g=1}^{2} w_g = 1$). The space Ω_1 has a one-dimensional normal probability density function $\mathcal{N}_1(\boldsymbol{x})$. On the other hand, the space Ω_2 has only one sample point. An F0 observation $\boldsymbol{o}$ consists of a continuous random variable $\boldsymbol{x}$ and a set of space indices X, that is,

$$\boldsymbol{o} = (X, \boldsymbol{x}) \tag{3.27}$$

where $X = \{1\}$ for voiced region and $X = \{2\}$ for unvoiced region. Then the observation probability of $\boldsymbol{o}$ is defined by

$$b(\boldsymbol{o}) = \sum_{g \in S(\boldsymbol{o})} w_g \mathcal{N}_g(V(\boldsymbol{o})) \tag{3.28}$$

where $V(\boldsymbol{o}) = \boldsymbol{x}$ and $S(\boldsymbol{o}) = X$. It is noted that, although $\mathcal{N}_2(\boldsymbol{x})$ does not exist for Ω_2, $\mathcal{N}_2(\boldsymbol{x}) \equiv 1$ is defined for simplicity of notation.

Using an HMM in which output probability in each state is given by Eq. (3.28), called MSD-HMM (Figure 3.2), voiced and unvoiced observations of F0 can be modeled in a unified model without any heuristic assumption [19]. Moreover, spectrum and F0 can be modeled simultaneously by

multi-stream MSD-HMM, in which spectral part is modeled by continuous probability distribution (CD), and F0 part is modeled by MSD (see Fig. 3.3). In the figure, $\boldsymbol{c}_t$, X_t^p, and $\boldsymbol{x}_t^p$ represent the spectral parameter vector, a set of space indices of F0, and F0 parameter at time t, respectively, and Δ and Δ^2 represent the delta and delta-delta parameters, respectively.

3.3 Decision-Tree-based Context Clustering

In continuous speech, parameter sequences of particular speech unit (e.g., phoneme) can vary according to phonetic context. To manage the variations appropriately, context dependent models, such as triphone/quinhpone models, are often employed. In the HMM-based speech synthesis system, we use more complicated speech units considering prosodic and linguistic context such as mora, accentual phrase, part of speech, breath group, and sentence information to model suprasegmental features in prosodic feature appropriately. However, it is impossible to prepare training data which cover all possible context dependent units, and there is great variation in the frequency of appearance of each context dependent unit. To alleviate these problems, a number of techniques are proposed to cluster HMM states and share model parameters among states in each cluster. Here, we describe a decision-tree-based state tying algorithm [10], [14], [22], [23]. This algorithm is often referred to as decision-tree-based context clustering algorithm.

3.3.1 Decision Tree

An example of a decision tree is shown in Fig. 3.4. The decision tree is a binary tree. Each node (except for leaf nodes) has a context related question, such as `R-silence?` ("is the previous phoneme a silence?") or `L-vowel?` ("is the next phoneme vowels?"), and two child nodes representing "yes" and "no" answers to the question. Leaf nodes have state output distributions. Using the decision-tree-based context clustering, model parameters of the speech units for the unseen contexts can be obtained, because any context reaches one of the leaf nodes, going down the tree starting from the root node then selecting the next node depending on the answer about the current context.

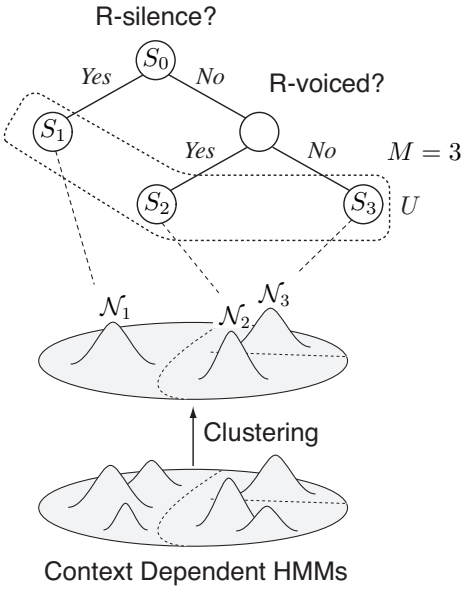


Figure 3.4: An example of decision tree.

3.3.2 Construction of Decision Tree

We will briefly review the construction method of the decision tree using the minimum description length (MDL) criterion [23]. Let S_0 be the root node of a decision tree and $U(S_1, S_2, \ldots, S_M)$ be a model defined for the leaf node set $\{S_1, S_2, \ldots, S_M\}$. Here, a model is a set of leaf nodes of a decision tree. A Gaussian pdf $\mathcal{N}_m$, which is obtained by combining several Gaussian pdfs classified into the node S_m, is assigned to each node S_m. An example of a decision tree for $M = 3$ is shown in Fig. 3.4. To reduce computational costs, we make the following three assumptions:

1. The transition probabilities of HMMs can be ignored in the calculation of the auxiliary function of the likelihood.

2. Context clustering does not change the frame or state alignment between the data and the model.

3. The auxiliary function of the log-likelihood for each state can be given by the sum of the log-likelihood for each data frame weighted by the state occupancy probability (Eq. 2.45) for each state.

From these assumptions, the auxiliary function $\mathcal{L}$ of the log-likelihood of the model U is given by

$$\mathcal{L}(U) \simeq \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\log\mathcal{N}_m(\boldsymbol{o}_t;\boldsymbol{\mu}_m,\boldsymbol{\Sigma}_m) \tag{3.29}$$

$$= \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\left(-\frac{(\boldsymbol{o}_t-\boldsymbol{\mu}_m)^\top\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_m)+L\log 2\pi+\log|\boldsymbol{\Sigma}_m|}{2}\right) \tag{3.30}$$

where $\boldsymbol{\mu}_m$ and $\boldsymbol{\Sigma}_m$ is the mean vector and the diagonal covariance matrix of the Gaussian pdf $\mathcal{N}_m$ at node S_m, respectively. If the re-estimation of the HMM parameters using EM algorithm (Eq. 2.43) was conducted fully, the estimated covariance matrix at convergence point is approximated by

$$\boldsymbol{\Sigma}_m = \frac{\sum_{t=1}^{T}\gamma_t(m)(\boldsymbol{o}_t-\boldsymbol{\mu}_m)(\boldsymbol{o}_t-\boldsymbol{\mu}_m)^\top}{\sum_{t=1}^{T}\gamma_t(m)}, \tag{3.31}$$

and furthermore since the covariance matrix is assumed to be diagonal,

$$\sum_{t=1}^{T}\gamma_t(m)(\boldsymbol{o}_t-\boldsymbol{\mu}_m)^\top\boldsymbol{\Sigma}_m^{-1}(\boldsymbol{o}_t-\boldsymbol{\mu}_m) = L\sum_{t=1}^{T}\gamma_t(m) \tag{3.32}$$

can be obtained. Thus, the auxiliary function $\mathcal{L}$ of the log-likelihood of the model U can be transformed as follows:

$$\mathcal{L}(U) \simeq -\frac{1}{2}\sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_t(m)\left(L+L\log 2\pi+\log|\boldsymbol{\Sigma}_m|\right). \tag{3.33}$$

Using Eq. (3.33), the description length [23] of the model U is given by

$$\mathcal{D}(U) \equiv -\mathcal{L}(U) + LM\log G + C \tag{3.34}$$

$$= \frac{1}{2}\sum_{m=1}^{M}\Gamma_m\left(L+L\log(2\pi)+\log|\boldsymbol{\Sigma}_m|\right) \tag{3.35}$$

$$+ LM\log G + C \tag{3.36}$$

where $\Gamma_m = \sum_{t=1}^{T}\gamma_t(m)$, $\gamma_t(m)$ is the state occupancy probability at node S_m, L is the dimensionality of the observation vector, $G = \sum_{m=1}^{M}\Gamma_m$, and C

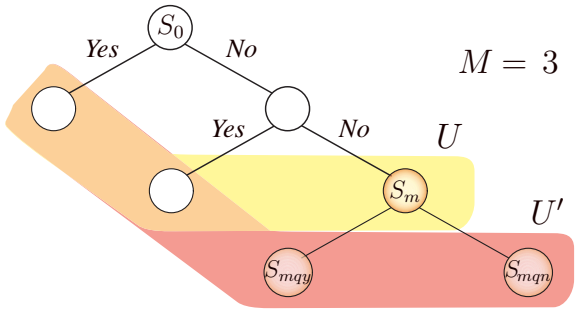


Figure 3.5: Splitting of node of decision tree.

is the code length required for choosing the model which is assumed here to be constant.

Suppose that node S_m of model U is split into two nodes, S_{mqy} and S_{mqn}, by using question q (Fig. 3.5). Let U' be the model obtained by splitting the S_m of model U by question q. The description length of model U' is calculated as follows:

$$\mathcal{D}(U') = \frac{1}{2}\Gamma_{mqy}\left(L + L\log(2\pi) + \log|\mathbf{\Sigma}_{mqy}|\right) \tag{3.37}$$

$$+ \frac{1}{2}\Gamma_{mqn}\left(L + L\log(2\pi) + \log|\mathbf{\Sigma}_{mqn}|\right) \tag{3.38}$$

$$+ \frac{1}{2}\sum_{\substack{m'=1 \\ m'\neq m}}^{M}\Gamma_{m'}\left(L + L\log(2\pi) + \log|\mathbf{\Sigma}_{m'}|\right) \tag{3.39}$$

$$+ L(M+1)\log G + C, \tag{3.40}$$

where the number of nodes of U' is $M+1$, Γ_{mqy}, Γ_{mqn} and $\mathbf{\Sigma}_{mqy}$, $\mathbf{\Sigma}_{mqn}$ are the state occupancy probabilities and the covariance matrices of Gaussian pdfs at nodes S_{mqy} and S_{mqn}, respectively. Hence, the difference between the description lengths before and after the splitting as follows:

$$\delta_m(q) = \mathcal{D}(U') - \mathcal{D}(U) \tag{3.41}$$

$$= \frac{1}{2}(\Gamma_{mqy}\log|\mathbf{\Sigma}_{mqy}| + \Gamma_{mqn}\log|\mathbf{\Sigma}_{mqn}| - \Gamma_m\log|\mathbf{\Sigma}_m|) \tag{3.42}$$

$$+ L\log G. \tag{3.43}$$

By using this difference, $\delta_m(q)$, we can automatically construct a decision tree. The process of constructing a decision tree is summarized below.

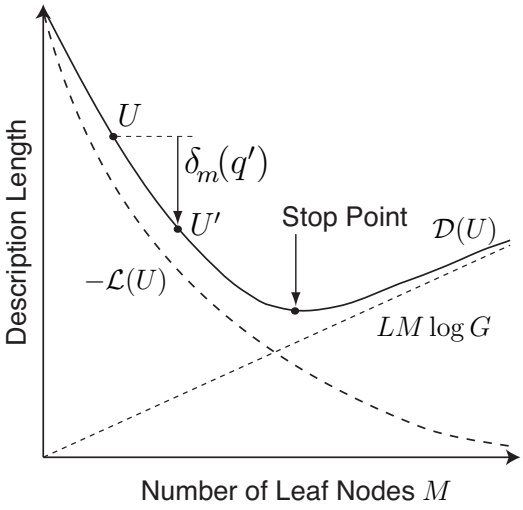


Figure 3.6: MDL-based decision-tree building.

1. Define initial model U as $U = \{S_0\}$.
2. Find node $S_{m'}$ in model U and question q' which minimize $\delta_{m'}(q')$.
3. Terminate if $\delta_{m'}(q') > 0$. If $\delta_{m'}(q') \leq 0$, stop the splitting of the nodes (Fig. 3.6).
4. Split node $S_{m'}$ by using question q' and replace U with the resultant node set.
5. Go to step 2.

3.4 HMM-based TTS System: Overview

A block-diagram of the HMM-based TTS system is shown in Fig. 3.7. The system consists of training stage and synthesis stage.

In the training stage, context dependent phoneme HMMs are trained using a speech database. Spectrum and F0 are extracted at each analysis frame as the static features from the speech database and modeled by multi-stream HMMs in which output distributions for the spectral and F0

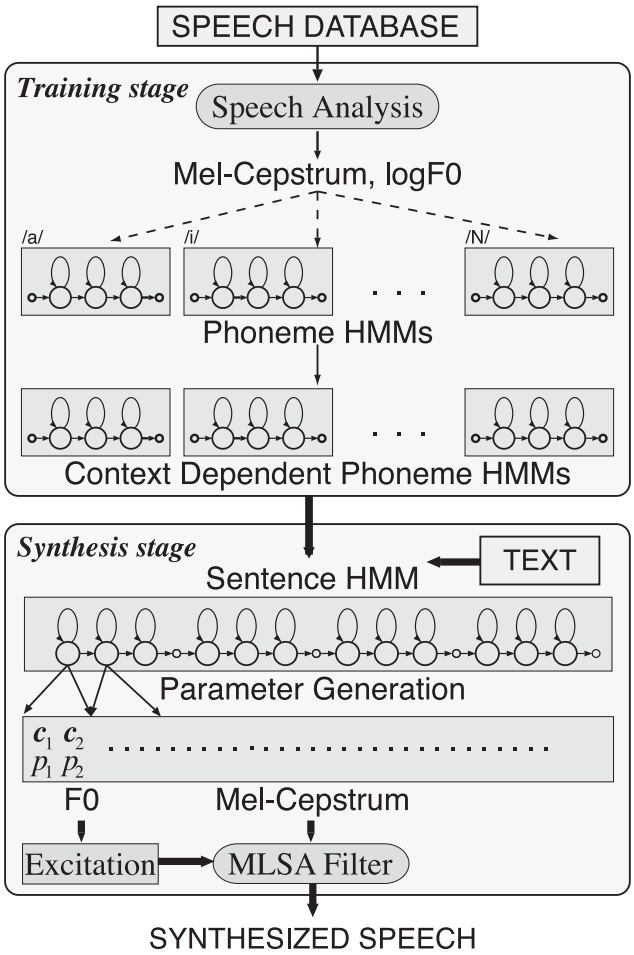


Figure 3.7: HMM-based speech synthesis system system.

parts are modeled using a continuous probability distribution and the multi-space probability distribution (MSD) [19], respectively. To model variations in the spectrum and F0, we take into account phonetic, prosodic, and linguistic contexts, such as phoneme identity contexts, stress-related contexts, and locational contexts. Then, the decision-tree-based context clustering technique [23], [24] is applied separately to the spectral and F0 parts of the context-dependent phoneme HMMs. In the clustering technique, a decision tree is automatically constructed based on the MDL criterion. We then perform re-estimation processes of the clustered context-dependent phoneme HMMs using the Baum-Welch (EM) algorithm. Finally, state durations are modeled by a multivariate Gaussian distribution [25], and the same state

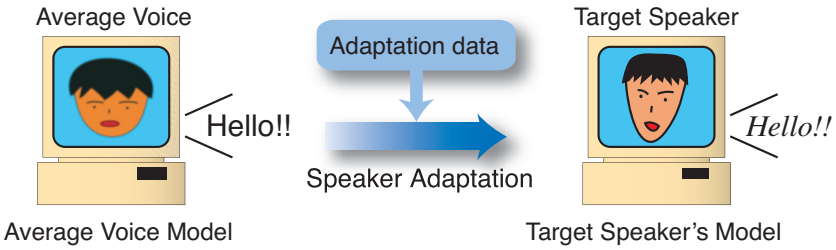


Figure 3.8: Speaker conversion

clustering technique is applied to the state duration models.

In the synthesis stage, first, an arbitrarily given text is transformed into a sequence of context-dependent phoneme labels. Based on the label sequence, a sentence HMM is constructed by concatenating context-dependent phoneme HMMs. From the sentence HMM, spectral and F0 parameter sequences are obtained based on the ML criterion [15] in which phoneme durations are determined using state duration distributions. Finally, by using an MLSA (Mel Log Spectral Approximation) filter [26] [27], speech is synthesized from the generated mel-cepstral and F_0 parameter sequences.

3.5 Speaker Conversion

In general, it is desirable that speech synthesis systems have the ability to synthesize speech with arbitrary speaker characteristics and speaking styles. For example, considering the speech translation systems which are used by a number of speakers simultaneously, it is necessary to reproduce input speakers' characteristics to make listeners possible to distinguish speakers of the translated speech. Another example is spoken dialog systems with multiple agents. For such systems, each agent should have his or her own speaker characteristics and speaking styles. From this point of view, a number of spectral/voice conversion techniques have been proposed [28]–[30]. However, speaker characteristics are also included in spectrum, fundamental frequency and duration parts [4], [5], and consequently, it is necessary to convert all these speech features to convert speech from one speaker to another.

In the HMM-based speech synthesis method, we can easily change spec-

tral and prosodic characteristics of synthetic speech by transforming HMM parameters appropriately since speech parameters used in the synthesis stage are statistically modeled by using the framework of the HMM. In fact, we have shown in [6]–[8], [31] that the TTS system can generate synthetic speech which closely resembles an arbitrarily given speaker's voice using a small amount of target speaker's speech data by applying speaker adaptation techniques such as MLLR (Maximum Likelihood Linear Regression) algorithm [32]. In the speaker adaptation, initial model parameters, such as mean vectors of output distributions, are adapted to a target speaker using a small amount of adaptation data uttered by the target speaker. The initial model can be speaker dependent or independent. For the case of speaker dependent initial model, since most of speaker adaptation techniques tend to work insufficiently between two speakers with significant difference in voice characteristics, it is required to select the speaker used for training the initial model appropriately depending on the target speaker. On the other hand, using speaker independent initial models, speaker adaptation techniques work well for most target speakers, though the performance will be lower than using speaker dependent initial models which matches the target speaker and has sufficient data. Since the synthetic speech generated from the speaker independent model can be considered to have averaged voice characteristics and prosodic features of speakers used for training, we refer to the speaker independent model as the "average voice model", and the synthetic speech generated from the average voice model as "average voice" (Fig. 3.8). In the next section, we will briefly describe the MLLR adaptation [32].

3.5.1 MLLR Adaptation

In the MLLR adaptation, which is the most popular linear regression adaptation, mean vectors of state output distributions for the target speaker's model are obtained by linearly transforming mean vectors of output distributions of the average voice model (Fig. 3.9),

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\zeta}\boldsymbol{\mu}_i + \boldsymbol{\epsilon}, \boldsymbol{\Sigma}_i) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{W}\boldsymbol{\xi}_i, \boldsymbol{\Sigma}_i) \tag{3.44}$$

where $\boldsymbol{\mu}_i$ are the mean vectors of output distributions for the average voice model. $\boldsymbol{W} = [\boldsymbol{\zeta}, \boldsymbol{\epsilon}]$ are $L \times (L+1)$ transformation matrices which transform

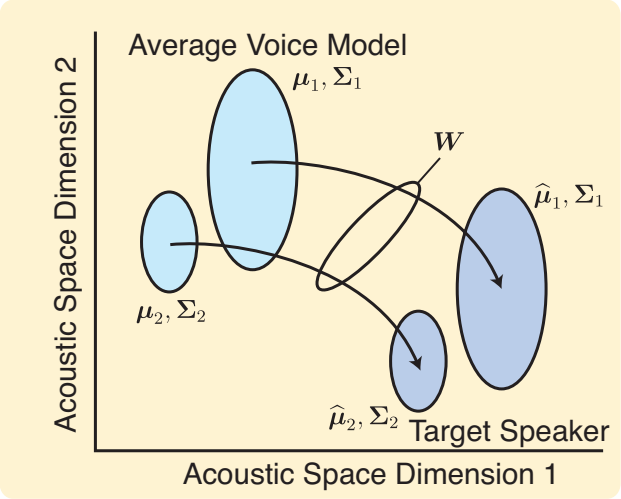


Figure 3.9: HMM-based MLLR adaptation algorithm.

average voice model into the target speaker for output distributions, and $\boldsymbol{\xi}_i = [\boldsymbol{\mu}_i^\top, 1]^\top$ are $(L+1)$-dimensional extended mean vectors. $\boldsymbol{\zeta}$ and $\boldsymbol{\epsilon}$ are $L \times L$ matrix and L-dimensional vector, respectively.

The MLLR adaptation estimates the transformation matrices $\boldsymbol{W}$ so as to maximize likelihood of adaptation data $\boldsymbol{O}$. The problem of the MLLR adaptation based on ML criterion can be expressed as follows:

$$\tilde{\boldsymbol{W}} = \underset{\boldsymbol{W}}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda, \boldsymbol{W}) \tag{3.45}$$

where λ is the parameter set of HMM. Re-estimation formulas based on Baum-Welch algorithm of the transformation matrices $\boldsymbol{W}$ can be derived as follows:

$$\overline{\boldsymbol{w}}_l = \boldsymbol{y}_l \boldsymbol{G}_l^{-1} \tag{3.46}$$

where $\boldsymbol{w}_l$ is the l-th row vector of $\boldsymbol{W}$, and $(L+1)$-dimensional vector $\boldsymbol{y}_l$, $(L+1) \times (L+1)$ matrix $\boldsymbol{G}_l$ are given by

$$\boldsymbol{y}_l = \sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t(r)\, \frac{1}{\Sigma_r(l)} o_t(l)\, \boldsymbol{\xi}_r^\top \tag{3.47}$$

$$\boldsymbol{G}_l = \sum_{r=1}^{R} \sum_{t=1}^{T} \gamma_t(r)\, \frac{1}{\Sigma_r(l)}\, \boldsymbol{\xi}_r \boldsymbol{\xi}_r^\top \tag{3.48}$$

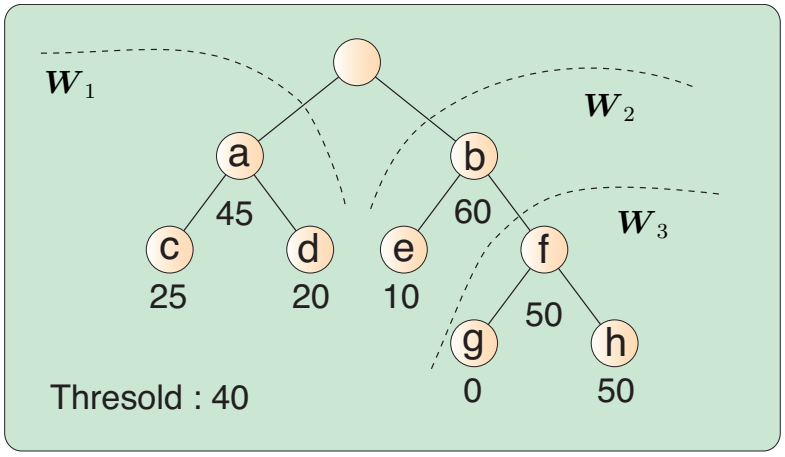


Figure 3.10: An example of the tying matrices based on the regression class tree.

where $\Sigma_r(l)$ is the l-th diagonal element of diagonal covariance matrix $\boldsymbol{\Sigma}_r$, and $o_t(l)$ is the l-th element of the observation vector $\boldsymbol{o}_t$. Note that $\boldsymbol{W}$ is tied across R distributions. When $1 \leq R < L$, we need to use generalized inverses with singular value decomposition.

Furthermore, we can straightforwardly apply this algorithm to the multi-space probability distribution (MSD) [7] for adapting F0 parameters to the target speaker. In the F0 adaptation of MSD-HMMs, only the mean vectors of distributions included in the voiced space are adapted. Therefore, only state occupancy counts for the voiced space are considered for tying the regression matrices.

3.5.2 Tying Transformation Matrices

In general, it is not always possible to estimate the MLLR transformation matrices $\boldsymbol{W}$ for every distribution, because the amount of adaptation data of a target speaker is small and generalized inverses method drastically decreases the accuracy of the transformation matrix. Therefore, we use tree structures to group the distributions in the model and to tie the transformation matrices in each group. In the tree structure, each node specifies a particular cluster of distributions in the model, and those nodes that have a

state occupancy count below a given threshold are placed in the same regression class as that of their parent node. The state occupancy count of node l is given by

$$s(l) = \sum_{r=1}^{R_l} \sum_{t=1}^{T} \gamma_t(r) \tag{3.49}$$

where R_l is the number of distributions belonging to node l. The tying makes it possible to adapt distributions which have no adaptation data. To determine the tying topology for the transformation matrices, regression class trees are constructed based on the distance between distributions (Fig. 3.10), such as a Euclidean distance measure [13].

Chapter 4

Shared-Decision-Tree-Based Context Clustering

In the decision-tree-based context clustering for "average voice model," which is a set of speaker independent speech synthesis units, the nodes of the decision tree do not always have training data of all speakers, and some nodes have data from only one speaker. This speaker-biased node causes degradation of quality of average voice and synthetic speech after speaker adaptation, especially in prosody. To overcome this problem, we propose a new context clustering technique, named "shared-decision-tree-based context clustering." An advantage of the technique is that every node of the decision tree always has the data of all speakers. In other words, there is no node lacking one or more training speakers' data. From the results of subjective tests, we show that the average voice models trained using the proposed technique can generate more natural sounding speech than the conventional average voice models.

4.1 Introduction

In chapter 3, we have described an HMM-based TTS system in which each speech synthesis unit is modeled by HMM. A distinctive feature of the system is that speech parameters used in the synthesis stage are generated directly from HMMs by using the parameter generation algorithm. Since the HMM-

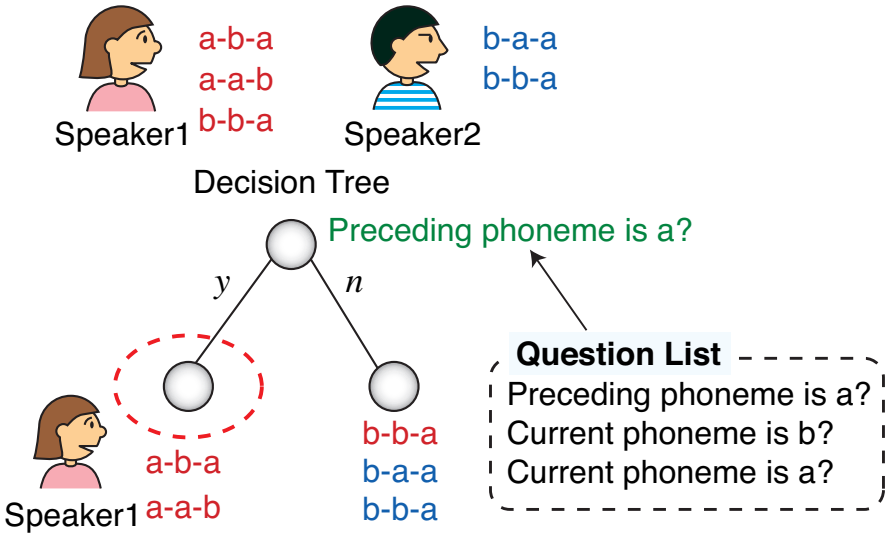


Figure 4.1: An example of speaker-biased clustering.

based TTS system uses HMMs as the speech units in both modeling and synthesis, we can easily change voice characteristics of synthetic speech by transforming HMM parameters appropriately. In fact, we have shown in [6]–[8] that the TTS system using the average voice model can generate synthetic speech which closely resembles an arbitrarily given speaker's voice using a small amount of target speaker's speech data by applying speaker adaptation techniques based on MLLR (Maximum Likelihood Linear Regression) algorithm [32]. In that system, quality of the average voice model crucially affects quality of synthetic speech generated from adapted models, and then training data including several speakers' speech data for the average voice model affects quality of the average voice model. To make the training data rich in phonetic and linguistic contexts and make the average voice model good quality, it is desirable that the individual sentence sets are used for respective speakers. However, synthetic speech generated from the average voice model trained using the individual sentence sets for respective speakers would sound unnatural compared to the model trained using the same sentence set for all speakers, especially when the trainng data of each training speaker differs widely. If the individual sentence sets are used for respective speakers, the contexts contained in each speaker's data are quite different.

As a result, after the decision tree based context clustering, the nodes of the tree do not always have training data of all speakers, and some nodes have data from only one speaker. Figure 4.1 shows the example of the biased clustering. In the figure, left child node of the root node is consisted of only speech data of training speaker A. This will cause degradation of quality of average voice, especially in prosody. For example, if the training speaker A has higher fundamental frequency than the training speaker B, the fundamental frequency of the average voice corresponding to the left child node becomes higher than that of the average voice corresponding to right child node. As a result, the unbalance node causes unnatural prosody.

To overcome this problem, in this chapter, we propose a new context clustering technique for the average voice model, which will be referred to as "shared decision tree context clustering" (STC). In the technique, we first train speaker dependent models using multi-speaker speech database, and construct a decision tree for context clustering common to these speaker dependent models. When a node of the decision tree is split, only the context related questions which are applicable to all speaker dependent models are adopted. As a result, every node of the decision tree always has the data of all speakers. Using the common decision tree, all speaker dependent models are clustered and an average voice model is obtained by combining Gaussian pdfs of speaker dependent models at each leaf node of the decision tree.

4.2 Shared-Decision-Tree-Based Context Clustering

4.2.1 Training of Average Voice Model

A block diagram of the training stage of average voice model using the proposing technique is shown in Fig. 4.2. First, context dependent models are separately trained for respective speakers to derive a decision tree common to these speaker dependent models for context clustering. Then, the decision tree, which we refer to as a shared decision tree, is constructed using an algorithm described in the next section from the speaker dependent models.

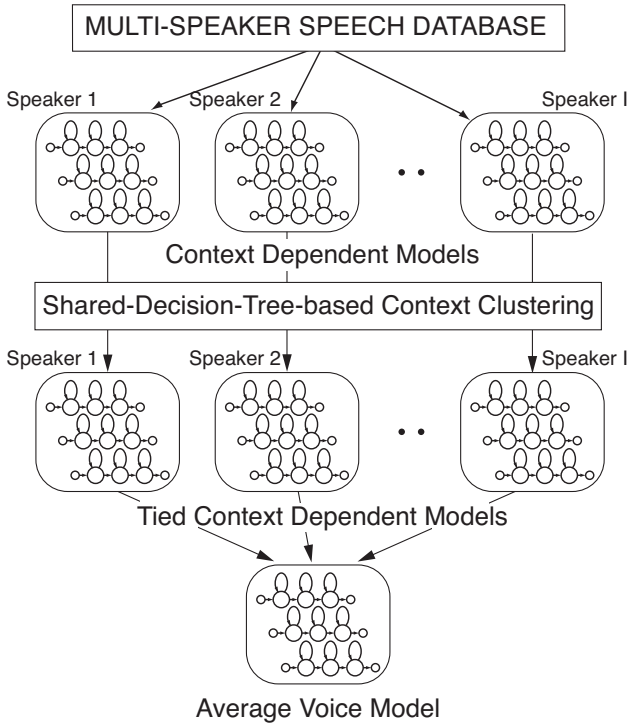


Figure 4.2: A block diagram of training stage of the average voice model.

Finally, all speaker dependent models are clustered using the shared decision tree. A Gaussian pdf of average voice model is obtained by combining all speakers' Gaussian pdfs at every node of the tree. After the re-estimation of parameters of the average voice model using training data of all speakers, state duration distributions is obtained for each speaker. Finally, state duration distributions of the average voice model is obtained by applying the same procedure.

4.2.2 Description Length of Average Voice Model

In the shared decision tree context clustering (STC), a speaker independent decision tree common to all speaker dependent models, namely a shared decision tree, is constructed based on the MDL criterion [23] in the same manner as the conventional decision-tree-based context clustering described in Sect. 3.3.

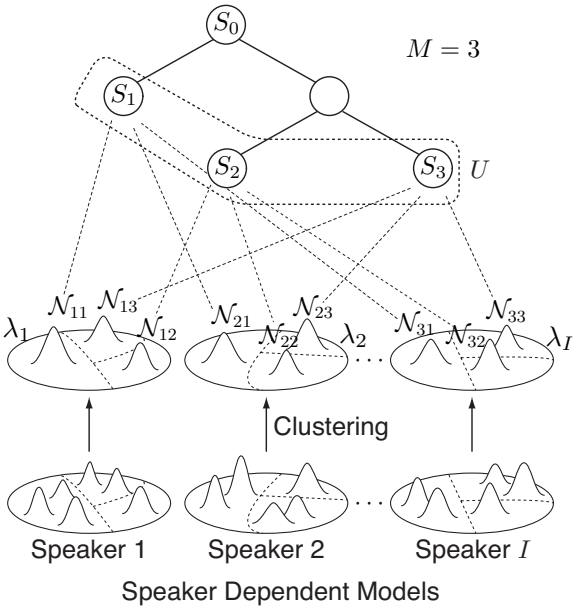


Figure 4.3: Context clustering for average voice model using a decision tree common to the speaker dependent models.

Let S_0 be the root node of a decision tree and $U(S_1, S_2, \ldots, S_M)$ be a model defined for the leaf node set $\{S_1, S_2, \ldots, S_M\}$ (see Fig. 4.3). A Gaussian pdf $\mathcal{N}_{im}$ of speaker i is assigned to each node S_m, and the set of Gaussian pdfs of each speaker i for the node set $\{S_1, S_2, \cdots, S_M\}$ is defined as $\lambda_i(S_1, S_2, \cdots, S_M) = \{\mathcal{N}_{i1}, \mathcal{N}_{i2}, \ldots, \mathcal{N}_{iM}\}$.

The description length of the speaker dependent model λ_i is given by

$$\begin{aligned}\mathcal{D}(\lambda_i) \equiv & -\mathcal{L}(\lambda_i) + cLM \log G_i + C \\ = & \frac{1}{2}\sum_{m=1}^{M} \Gamma_{im} \left(L + L\log(2\pi) + \log|\mathbf{\Sigma}_{im}|\right) \\ & + cLM \log G_i + C, \end{aligned} \tag{4.1}$$

where L is the dimensionality of the data vector, Γ_{im} and $\mathbf{\Sigma}_{im}$ are the state occupancy count and the covariance matrix of Gaussian pdf of speaker i at node S_m, respectively. $G_i = \sum_{m=1}^{M} \Gamma_{im}$, and C is the code length required for choosing the model which is assumed here to be constant. Note that we introduce a weight c for adjusting the model size. As the weighting factor c decreases, the number of leaf nodes increases, and vice versa (see Fig. 4.4.).

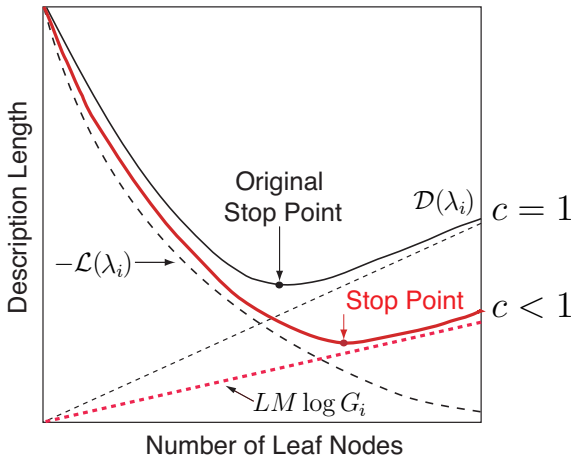


Figure 4.4: MDL criterion and the weight factor c.

We now define the description length for the model U as

$$\widehat{\mathcal{D}}(U) \equiv \sum_{i=1}^{I} \mathcal{D}(\lambda_i), \tag{4.2}$$

where I is the total number of speakers. Suppose that node S_m of model U is split into two nodes by applying a question q. Let U' be the model obtained by splitting S_m of model U by the question q. Then we define the difference between the description lengths after and before the splitting as follows:

$$\delta_m(q) = \widehat{\mathcal{D}}(U') - \widehat{\mathcal{D}}(U). \tag{4.3}$$

The procedure of construction of the shared decision tree is the same as that of construction of the conventional decision tree described in Sect. 3.3 except that only the questions which are applicable to all speaker dependent models are adopted in step 2. Figure 4.5 shows the example. In the figure, the question "Is preceding phoneme a?" is not applicable to the training speaker A. Hence, we eliminate the question from the question list for the node splitting. As a result, every node of the shared decision tree always has training data from all training speakers.

After the construction of the shared decision tree, we obtain Gaussian pdfs of the average voice model by combining Gaussian pdfs of speaker dependent models. The mean vector $\boldsymbol{\mu}_m$ and the covariance matrix $\boldsymbol{\Sigma}_m$ of the Gaussian

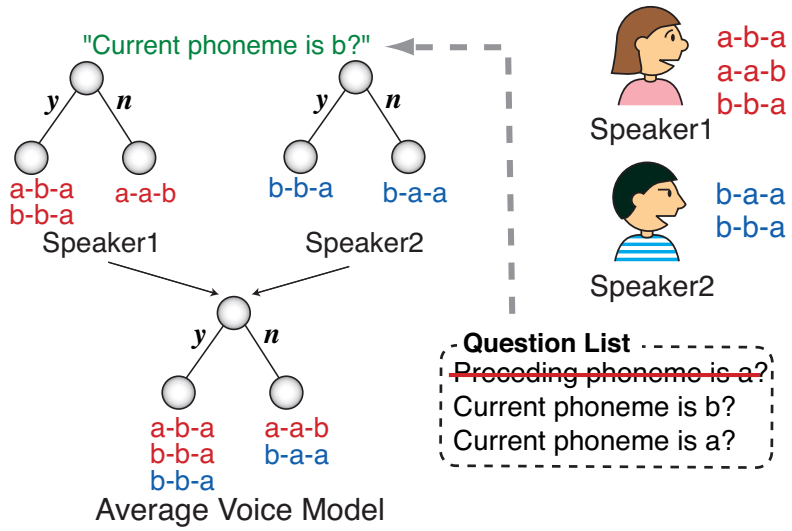


Figure 4.5: Questions which are applicable to all training speakers are only adopted in node splitting.

pdf at node S_m are calculated as follows:

$$\boldsymbol{\mu}_m = \frac{\sum_{i=1}^{I} \Gamma_{im} \boldsymbol{\mu}_{im}}{\sum_{i=1}^{I} \Gamma_{im}}, \tag{4.4}$$

$$\boldsymbol{\Sigma}_m = \frac{\sum_{i=1}^{I} \Gamma_{im} \left(\boldsymbol{\Sigma}_{im} + \boldsymbol{\mu}_{im} \boldsymbol{\mu}_{im}^{\top} \right)}{\sum_{i=1}^{I} \Gamma_{im}} - \boldsymbol{\mu}_m \boldsymbol{\mu}_m^{\top}, \tag{4.5}$$

where $\cdot^{\top}$ denotes matrix transpose, and Γ_{im} and $\boldsymbol{\mu}_{im}$ are the state occupancy count and the mean vector of the Gaussian pdf of speaker i at node S_m, respectively.

4.3 Experiments

4.3.1 Experimental Conditions

We used 503 phonetically balanced sentences from ATR Japanese speech database (Set B) [1] for training HMMs. Based on phoneme labels and linguistic information included in the database, we made context dependent

[1] http://www.red.atr.co.jp/database_page/digdb.html

phoneme labels. We used 42 phonemes including silence and pause as shown in Table 4.1 and took the following phonetic and linguistic contexts into account:

- the number of morae in a sentence;
- the position of the breath group in a sentence;
- the number of morae in the {preceding, current, and succeeding} breath groups;
- the position of the current accentual phrase in the current breath group;
- the number of morae and the type of accent in the {preceding, current, and succeeding} accentual phrases;
- the part of speech of the {preceding, current, and succeeding} morpheme;
- the position of the current mora in the current accentual phrase;
- the differences between the position of the current mora and the type of accent;
- {preceding, current, and succeeding} phonemes.

It is noted that a unit of position is mora.

Speech signals were sampled at a rate of 16kHz and windowed by a 25ms Blackman window with a 5ms shift. Then mel-cepstral coefficients were obtained by mel-cepstral analysis [27], [33]. F0 values were extracted using ESPS `get_F0` program [34]. The feature vectors consisted of 25 mel-cepstral coefficients including the zeroth coefficient, logarithm of fundamental frequency, and their delta and delta-delta coefficients.

The context dependent phoneme HMMs are 5-state left-to-right models. The average voice models were trained using from 50 to 300 sentences of each speaker's speech data. Speakers were 3 females (FKN, FKS, FYM) and 3 males (MHO, MHT, MYI). Table 4.2 (a) and (b) show the number of sentences per speaker and corresponding sentence sets used for training. The

Table 4.1: Phonemes list used in the experiments.

	Voiced	Unvoiced
Vowel	a i u e o	A I U E O
Plosive	b by d dy g gy	k ky p py f t
Fricative	z j	h hy s sh
Affricate		ts ch
Liquid	r ry	
Nasal	m N my n ny	
Semi-vowel	y w	
Double consonant	cl	
Silence / Pause	sil pau	

sentence sets A–I consist of 50 sentences, respectively. Table 4.2 (a) shows the case in which the same sentence sets were used for all speakers, and Table 4.2 (b) shows the case in which the individual sentence sets were used for respective speakers. A model trained using 450 sentences (sentence set A–I) per speaker is used as a reference model of the subjective evaluations in Sec. 4.3.4. Average voice models are trained using the conventional technique [7], [8] described in Sec. 3.3.2 and the proposed technique (described in Sec. 4.2). In the proposed technique, the total number of parameters of all speaker dependent models is considered, while the number of parameters of only one speaker independent model is considered in the conventional technique. This causes increase of the last term on the right-hand side of (4.1), and results in a higher node splitting threshold for the increase in log-likelihood than the conventional technique. Consequently, if the weighting factor c of the description length of the proposed technique is set to unity, as in the conventional technique, the number of the leaf nodes of the proposed decision tree becomes considerably small. The considerable decrease of the leaf nodes of the decision tree makes the synthetic speech unnatural. Therefore, we adjust the weighting factor c to increase the number of leaf nodes of the proposed decision tree. From the results of preliminary experiments, we set the weighting factor c of the description length to 1 for the conventional

Table 4.2: Sentences per speaker and sentence sets used for training.

(a) The same sentence sets.

Sentences per Speaker	Female			Male		
	FKN	FKS	FYM	MHO	MHT	MYI
50	A	A	A	A	A	A
100	A,B	A,B	A,B	A,B	A,B	A,B
150	A-C	A-C	A-C	A-C	A-C	A-C
200	A-D	A-D	A-D	A-D	A-D	A-D
250	A-E	A-E	A-E	A-E	A-E	A-E
300	A-F	A-F	A-F	A-F	A-F	A-F

(b) The individual sentence sets.

Sentences per Speaker	Female			Male		
	FKN	FKS	FYM	MHO	MHT	MYI
50	A	B	C	D	E	F
100	A,B	B,C	C,D	D,E	E,F	F,G
150	A-C	B-D	C-E	D-F	E-G	F-H
200	A-D	B-E	C-F	D-G	E-H	F-I
250	A-E	B-F	C-G	D-H	E-I	A,F-I
300	A-F	B-G	C-H	D-I	A,E-I	A,B,F-I

models and 0.4 for the proposed models, respectively.

4.3.2 Results of Context Clustering

Table 4.3 (a) and (b) show the number of leaf nodes of the decision trees constructed using the conventional and proposed techniques. Table 4.3 (a) shows the result for the case of the same sentence sets, and Table 4.3 (b) shows the results for the case of the individual sentence sets. The numbers of distributions of the reference model using 450 sentences (sentence set A–I) per speaker are 1697, 2887, and 2892 for spectrum, F0, and state duration, respectively.

Table 4.4 shows the number of leaf nodes which did not have training

Table 4.3: The number of leaf nodes of decision trees.

(a) The same sentence sets.

Sentences per Speaker	Conventional			Proposed		
	Spec.	F0	Dur.	Spec.	F0	Dur.
50	405	690	584	608	907	811
100	622	1126	934	998	1597	1372
150	799	1418	1287	1339	1605	1825
200	1004	1794	1660	1599	2578	2230
250	1163	2060	1923	1895	2977	2754
300	1310	2270	2271	2138	3433	3098

(b) The individual sentence sets.

Sentences per Speaker	Conventional			Proposed		
	Spec.	F0	Dur.	Spec.	F0	Dur.
50	419	1011	911	548	818	814
100	670	1674	1416	913	1416	1497
150	834	2026	1820	1252	2009	2073
200	1015	2261	1974	1504	2438	2502
250	1158	2419	2293	1779	2908	2931
300	1284	2472	2369	2002	3257	3203

data of all speakers and its percentage when the average voice models were trained using the individual 50-sentence sets of each speaker. In Table 4.4, (A) shows the number of leaf nodes lacking one or more speakers' data and its percentage, and (B) shows the number of leaf nodes which had only one speaker's data and its percentage. From Table 4.4, it can be seen that 50% of leaf nodes of the conventional decision tree for F0 lacked one or more speakers' data and 19% of leaf nodes had only one speaker's data. On the other hand, theoretically, every leaf node of the proposed decision tree has the training data of all speakers. Therefore, there is no node lacking one or more speakers' data.

Figure 4.6 and 4.7 show examples of generated F0 contours for a Japanese sentence /he-ya-i-ppa-i-ni-ta-ba-ko-no-no-mu-ga-ta-chi-ko-me- pau-yu-ru-ya-

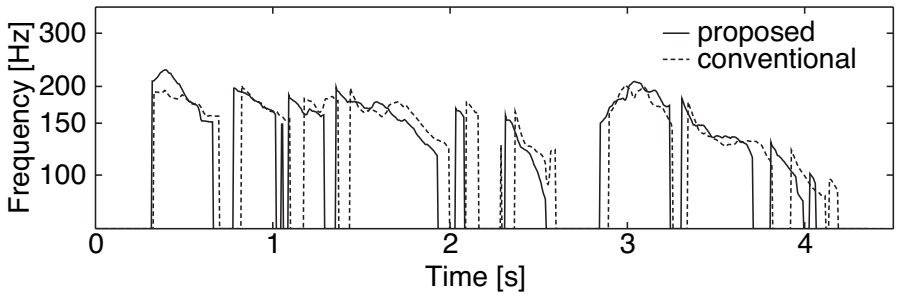
Table 4.4: The number of leaf nodes which did not have training data of all speakers. (A) shows the number of leaf nodes lacking one or more speakers' data and its percentage. (B) shows the number of leaf nodes which had only one speaker's data and its percentage.

	Conventional		Proposed	
	(A)	(B)	(A)	(B)
Spectrum	37 (8%)	14 (3%)	0 (0%)	0 (0%)
F0	505 (50%)	197 (19%)	0 (0%)	0 (0%)

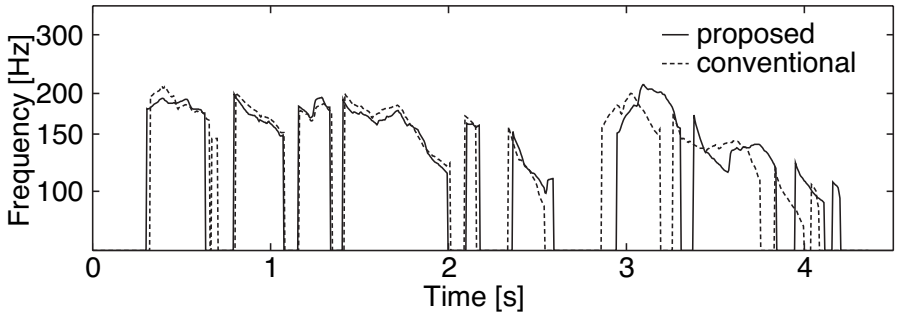
ka-ni-u-go-i-te-i-ru/ (meaning "Cigarette smoke fills the whole room, and is moving gently," in English) which is not included in the training sentences. Figure 4.6 shows the result for the case of the same sentence sets, and Fig. 4.7 shows the results for the case of the individual sentence sets. In Fig. 4.6 and Fig. 4.7, (a) and (b) show the F0 contours generated from the average voice models trained using 50 sentences and 300 sentences per speaker. Dotted line and solid line show the F0 contours generated from the average voice models clustered using conventional and proposed techniques, respectively.

From Fig. 4.6, it can be seen that the conventional and proposed techniques provide similar results when the same sentence sets were used. This is because the intersection of context sets contained in the respective speakers' training data are large when the sentence sets were the same[2]. On the other hand, from Fig. 4.7 (a), we can see that the F0 contours generated from the conventional and proposed models are quite different at the beginning of the sentence; the values of F0 generated from the conventional model are unnaturally high, whereas there is no obviously unnatural part in the F0 contour generated from the proposed model. This is due to the fact that leaf nodes of the conventional model corresponding to the beginning of the sentence had only one female speaker's training data. However, from Fig. 4.7 (b), we can see that there is no significant difference between the F0 contours generated

[2]The context sets of respective speakers' data do not always the same even if the sentence sets are the same, since some contextual factors, such as position of pause and accentual type, are not determined by text and vary depending on speakers.



(a) 50 sentences per speaker



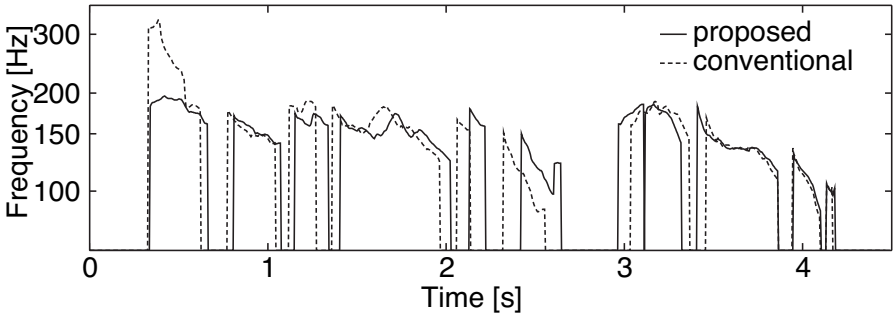
(b) 300 sentences per speaker

Figure 4.6: Comparison of F0 contours generated from average voice models constructed using conventional and proposed techniques for the same sentence sets.

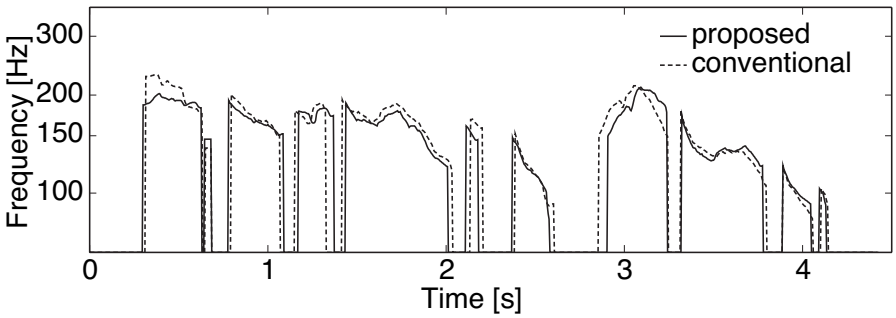
from the conventional and proposed models. This is due to the fact that the size of the intersection of sentence sets increases as the number of sentences for each speaker increases. For example, when the number of sentences for each speaker is 300, the sentence set F is included by all speakers' sentence sets. As a result, the number of leaf nodes biased to a speaker or a gender decreases in the conventional model.

4.3.3 Subjective Evaluations

We conducted paired comparison tests for synthetic speech generated from the average voice models trained using the conventional and proposed techniques. Subjects were eleven males. For each subject, eight test sentences were chosen at random from the 53 test sentences which were not contained in the training data. Subjects were presented a pair of average voices syn-



(a) 50 sentences per speaker



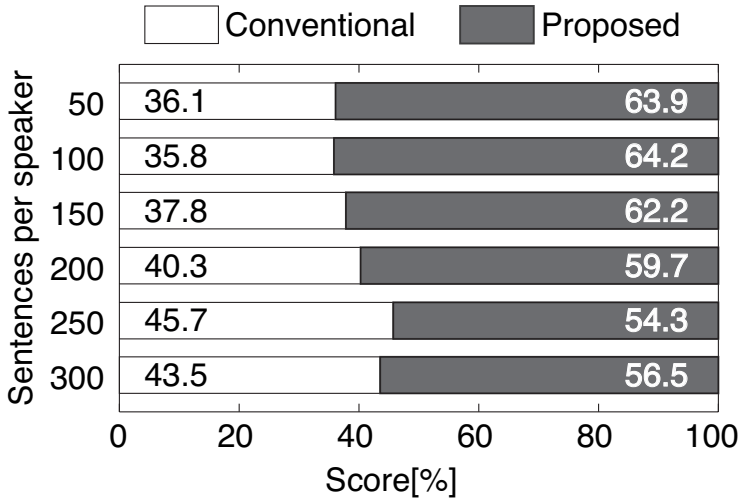
(b) 300 sentences per speaker

Figure 4.7: Comparison of F0 contours generated from average voice models constructed using conventional and proposed techniques for the individual sentence sets.

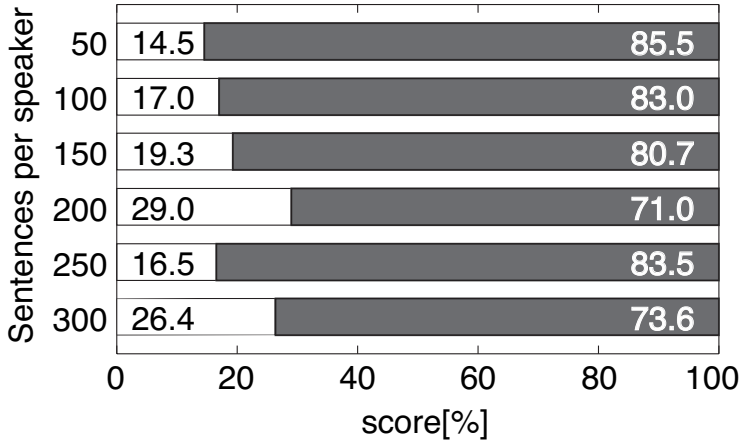
thesized from average voice models trained using conventional and proposed techniques in random order, and asked which synthetic speech sounded more natural.

Figure 4.8 shows the results of the paired comparison test. In Fig. 4.8, (a) shows the results for the case of the same sentence sets, and (b) shows the results for the case of the individual sentence sets. The horizontal axes indicate the preference score, and the bars indicate the results for the models trained using 50, 100, 150, 200, 250 and 300 sentences per speaker, respectively.

From these figures, it can be seen that the average voice generated from the proposed models sound more natural than the average voice from conventional models regardless of the number of training sentences and sentence sets. It can also be seen that differences between the scores of proposed and conventional models are greater in the case of the individual sentence sets



(a) The same sentence sets.



(b) The individual sentence sets.

Figure 4.8: Result of the paired comparison test.

for respective speakers than the case of the same sentence sets. Moreover, the difference becomes greater as the number of training sentences decreases. Especially, when individual sentence set for respective training speakers were used and the number of sentences for each speaker is less than 150, the scores of the proposed technique attained more than 80%.

This can be due to the following reason. When the sentence sets for respective training speakers are different, context sets of respective speakers' data become quite different, and the intersection of context sets becomes

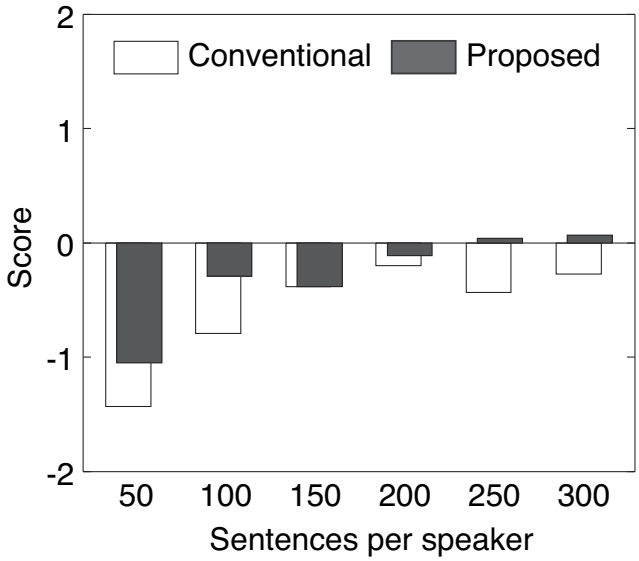
smaller as the number of sentences decreases. Even if the same sentence set is used for all speakers, the context sets of respective speakers' data are not usually identical. Using the conventional technique, as the context sets of respective speakers' data becomes more different, the number of leaf nodes lacking one or more speakers' data increases, and quality of average voice generated from conventional models tends to degrade. On the other hand, since the proposed technique is robust to difference of context sets between training speakers' data, quality of average voice generated from proposed models does not degrade seriously.

4.3.4 Comparison of Number of Training Data

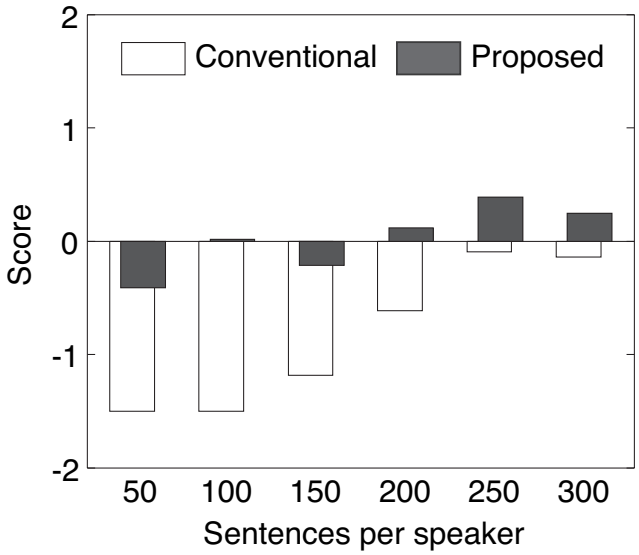
We conducted a comparison category rating test to evaluate the naturalness of the average voice generated from the model trained using the proposed technique. As a reference model, we used a conventional model trained using 450 sentences (sentence set A–I) per a speaker. A subject was required to judge quality of test speech on a seven point scale (3: much better, 2: better, 1: slightly better, 0: almost the same, -1: slightly worse, -2: worse, -3: much worse) compared to reference speech on naturalness and intelligibility. Subjects were seven males. For each subject, four test sentences were chosen at random from the 53 test sentences which were not contained in training data.

Figure 4.9 shows the result of the evaluation of the naturalness. In Fig. 4.9, (a) shows the results for the case of the same sentence sets, and (b) shows the results for the case of the individual sentence sets. The vertical axes indicate the average score and the horizontal axes indicate the number of sentences.

From this figure, it is seen that naturalness of the average voice of the proposed technique is higher than the conventional technique. Comparing Fig. 4.9 (a) and (b), when the number of sentences is limited, scores for the conventional models using the individual sentence sets are lower than the conventional models using the same sentence sets, whereas scores for proposed models using the individual sentence sets are higher than proposed models using same sentence sets. Moreover, using proposed technique and



(a) The same sentence sets.



(b) The individual sentence sets.

Figure 4.9: Result of evaluation of naturalness.

the individual sentence sets, there is only a little degradation on naturalness of the average voice even when training data is limited. In fact, the average voice which was trained using only 50 sentences per speaker is almost equivalent on naturalness to the average voice trained using 450 sentences by

Table 4.5: Result of the evaluation of average voice model trained using speech data with F0 normalization. Score shows the average number of sentences which are judged to be clearly unnatural.

Clustering Method	Conventional		Proposed	
F0 Normalization	No	Yes	No	Yes
Score	21.0	14.2	7.0	1.3

the conventional technique. This is due to the facts that difference between context sets does not cause degradation of naturalness of average voice for proposed models, and that when the size of database is almost the same, the average voice model trained using "context-rich" database can generate more natural sounding speech than model trained using "context-poor" database.

4.3.5 Evaluations of The Model with F0 Normalization

To show effectiveness of the proposed technique, we compared it with an F0 normalization technique. F0 normalization was achieved by shifting F0 contours in logarithmic domain so that the mean value of F0 of each speaker is equal to mean value of F0 of all training speakers. Then average voice models were trained using individual 50-sentence sets. Subjects were five males and required to judge whether or not test speech was clearly unnatural. The test sentences were 53 sentences which were not contained in the training data.

Table 4.5 shows the result of the evaluation. In the table, each score shows the average number of sentences which are judged to be clearly unnatural. It can be seen that the average voices using the training data with F0 normalization sound more natural than those without F0 normalization. It is due to the fact that the influence of leaf nodes biased to a speaker or a gender is reduced in the decision tree of F0. It can also be seen that the average voices using the proposed technique sound more natural than the conventional technique with the F0 normalization. It has been observed from the informal listening tests that the proposed technique reduces the influence of leaf nodes biased to a speaker or a gender in the decision tree of spectrum

and state duration, as well as F0.

4.4 Conclusion

In this paper, we have proposed a new context clustering technique, named shared decision tree context clustering, for an HMM-based speech synthesis system. An advantage of the technique is that every node of the decision tree always has the data of all speakers. In other words, there is no node lacking one or more speakers' data. We have shown that the average voice models constructed using the proposed technique can synthesize more natural sounding speech than the conventional models.

Future work will focus on evaluation of synthetic speech generated using models adapted from average voice models based on the proposed technique. Training using the proposed technique and SAT (Speaker Adaptive Training) [35] at the same time is also our future work.

Chapter 5

Speaker Adaptive Trainng

This chapter describes a new training method of average voice model for speech synthesis. When trainng data of each training speaker differs widely, the distributions of average voice model often have bias depending on speaker and/or gender and this will degrade the quality of synthetic speech. In the proposed method, to reduce the influence of speaker dependence, we incorporate speaker adaptive training and shared decision tree context clustering into the training procedure of average voice model. From the results of subjective tests, we show that the average voice model trained using the proposed method generates more natural sounding speech than the conventional average voice model. Moreover, it is shown that voice characteristics and prosodic features of synthetic speech generated from the adapted model using the proposed method are closer to the target speaker than the conventional method.

5.1 Introduction

To obtain higher performance in the speaker adaptation to a wide variety of target speakers, the initial model of the adaptation, namely the average voice model, should not have any bias depending on speaker and/or gender. However, it would occur that the distributions of the average voice model have relatively large bias depending on speaker and/or gender included in the training speech database, especially when trainng data of each training

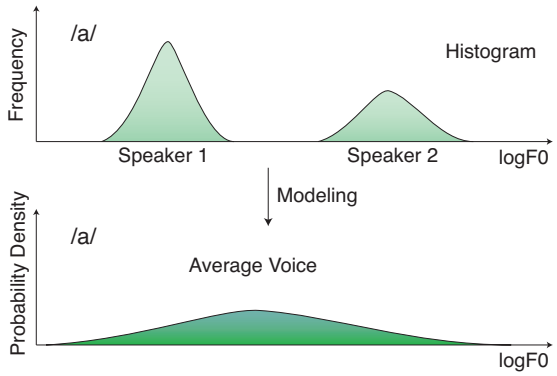


Figure 5.1: Speaker independent training.

speaker differs widely. This will affect model adaptation performance and degrade the quality of synthetic speech. To overcome this problem, we proposed a technique for constructing a decision tree used for clustering the average voice model [36]. Using this technique, which we call "shared decision tree context clustering (STC)" , every node of the decision tree always has training data from all speakers included in the training speech database. As a result, each distribution of the average voice model reflects the statistics of all speakers. Moreover, it has been shown that the quality of the average voice is improved by using this technique [36].

In this paper, we propose a new training method of average voice model for further reducing influence of speaker dependence and improving the quality of both average voice and synthetic speech of the given target speaker. In the proposing method, we incorporate speaker adaptive training (SAT) [35] as well as STC into the training procedure of the average voice model. Specifically, STC is used for clustering distributions of spectrum, fundamental frequency (F0), and state duration, then SAT is used for re-estimation of parameters of spectrum and F0.

5.2 Speaker Adaptive Training

The training data for the average voice model is consisted of several training speakers' speech data. If we apply the normal model training method to the average voice model directly, the model parameters of the average voice model

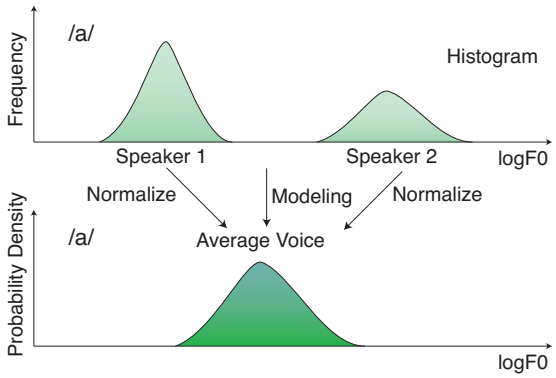


Figure 5.2: Speaker adaptive training.

are affected by the influence of speaker differences of the training speakers. Figure 5.1 shows the example. In the figure, two distributions of logarithm of fundamental frequency for two training speakers are described. If we apply normal model training method directly, the difference among the trainng speakers affect the distribution for average voice model and thereby we obtain a distribution having wide variance and dependence on training speakers (Fig. 5.1 (a)). To obtain higher performance in the speaker adaptation to a wide variety of target speakers, we should normalize the large diffenrence and dependence on speakers and/or gender (Fig. 5.1 (b)). Speaker adaptive training (SAT) [35] is a kind of the speaker normalization algorithm for normalizing the influence of the large diffenrence and the speaker dependence among the training speakers.

In the speake adaptive training [35], the speaker difference between training speaker's voice and canonical average voice is assumed to be expressed as a simple linear regression function of mean vectors of state output distributions

$$\boldsymbol{\mu}_i^{(f)} = \boldsymbol{\zeta}^{(f)}\boldsymbol{\mu}_i + \boldsymbol{\epsilon}^{(f)} = \boldsymbol{W}^{(f)}\boldsymbol{\xi}_i \tag{5.1}$$

where $\boldsymbol{\mu}_i^{(f)}$ and $\boldsymbol{\mu}_i$ is the mean vectors of state output distributions for training speaker f and the average voice model, respectively. $\boldsymbol{W}^{(f)} = \left[\boldsymbol{\zeta}^{(f)}, \boldsymbol{\epsilon}^{(f)}\right]$ is transformation matrices which indicate the speaker difference between training speaker f and average voice in state output distributions. After the estimating the transformation matrices for each training speaker, a canonical/average voice model is estimated so that the training speaker's model

transformed by the matrices maximizes the likelihood for the training data of the training speakers.

Let F be the total number of the training speakers, $\boldsymbol{O} = \{\boldsymbol{O}^{(1)}, \cdots, \boldsymbol{O}^{(F)}\}$ be all training data, and $\boldsymbol{O}^{(f)} = \{\boldsymbol{o}_{1_f}, \cdots, \boldsymbol{o}_{T_f}\}$ be the training data of length T_f for speaker f. The HMM-based speaker adaptive training simultaneously estimates the parameter set of HMM λ and the transformation matrix $\boldsymbol{W}^{(f)}$ for each training speaker so as to maximize likelihood of the training data $\boldsymbol{O}$. The problem of the HMM-based speaker adaptive training based on ML criterion can be formulated as follows:

$$\begin{aligned}(\tilde{\lambda}, \tilde{\Lambda}) &= \underset{\lambda,\Lambda}{\operatorname{argmax}} P(\boldsymbol{O}|\lambda, \Lambda) \\ &= \underset{\lambda,\Lambda}{\operatorname{argmax}} \prod_{f=1}^{F} P(\boldsymbol{O}^{(f)}|\lambda, \Lambda^{(f)})\end{aligned} \tag{5.2}$$

where $\Lambda = (\boldsymbol{W}^{(1)}, \cdots, \boldsymbol{W}^{(F)})$ is a set of the transformation matrices for the training speakers. Here we use a three-step iterative procedure to update the parameters [35]. First, we estimate the transformation matrices Λ while keeping λ fixed to the current values. The re-estimation formulas based on Baum-Welch algorithm of the parameter set Λ are identical to Eq. (3.45). We then estimate the mean vectors of λ using the updated transformation matrices while keeping the covariance matrices of λ fixed to the current values. Finally, the covariance matrices of λ are estimated using the updated transformation matrices and the updated mean vectors. The re-estimation formulas of the parameter set λ are given by

$$\begin{aligned}\overline{\boldsymbol{\mu}}_i = &\left[\sum_{f=1}^{F}\sum_{t=1}^{T_f} \gamma_t(i)\ \overline{\boldsymbol{\zeta}}^{(f)^\top} \boldsymbol{\Sigma}_i^{-1} \overline{\boldsymbol{\zeta}}^{(f)}\right]^{-1} \cdot \\ &\left[\sum_{f=1}^{F}\sum_{t=1}^{T_f} \gamma_t(i)\, \overline{\boldsymbol{\zeta}}^{(f)^\top} \boldsymbol{\Sigma}_i^{-1} (\boldsymbol{o}_{t_f} - \overline{\boldsymbol{\epsilon}}^{(f)})\right]\end{aligned} \tag{5.3}$$

$$\overline{\boldsymbol{\Sigma}}_i = \frac{\displaystyle\sum_{f=1}^{F}\sum_{t=1}^{T_f} \gamma_t(i)(\boldsymbol{o}_{t_f} - \overline{\boldsymbol{\mu}}_i^{(f)})(\boldsymbol{o}_{t_f} - \overline{\boldsymbol{\mu}}_i^{(f)})^\top}{\displaystyle\sum_{f=1}^{F}\sum_{t=1}^{T_f} \gamma_t(i)} \tag{5.4}$$

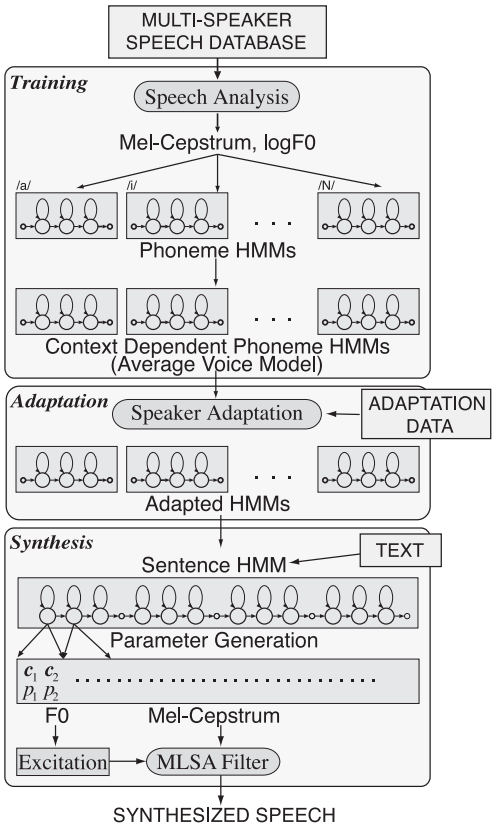


Figure 5.3: A block diagram of an HMM-based speech synthesis system using the average voice model and speaker adaptation.

where $\overline{\boldsymbol{\mu}}_i^{(f)} = \overline{\boldsymbol{\zeta}}^{(f)}\overline{\boldsymbol{\mu}}_i + \overline{\boldsymbol{\epsilon}}^{(f)}$ is the mean vector transformed into the training speaker f using the updated mean vector and the updated transformation matrix. Here $\overline{\boldsymbol{W}}^{(f)} = \left[\overline{\boldsymbol{\zeta}}^{(f)}, \overline{\boldsymbol{\epsilon}}^{(f)}\right]$ is the updated transformation matrix for the training speaker f. This adaptive training algorithm can be also viewed as a generalized algorithm of cepstral mean normalization (CMN), vocal tract length normalization (VTLN) [37]–[39], or F0 shift normalization used in Sect. 4.3.5.

5.3 TTS System Using Speaker Adaptation

Speech synthesis system using speaker adaptation for generating arbitrary speaker's voice characteristics is described in detail in [7], [8]. In this section,

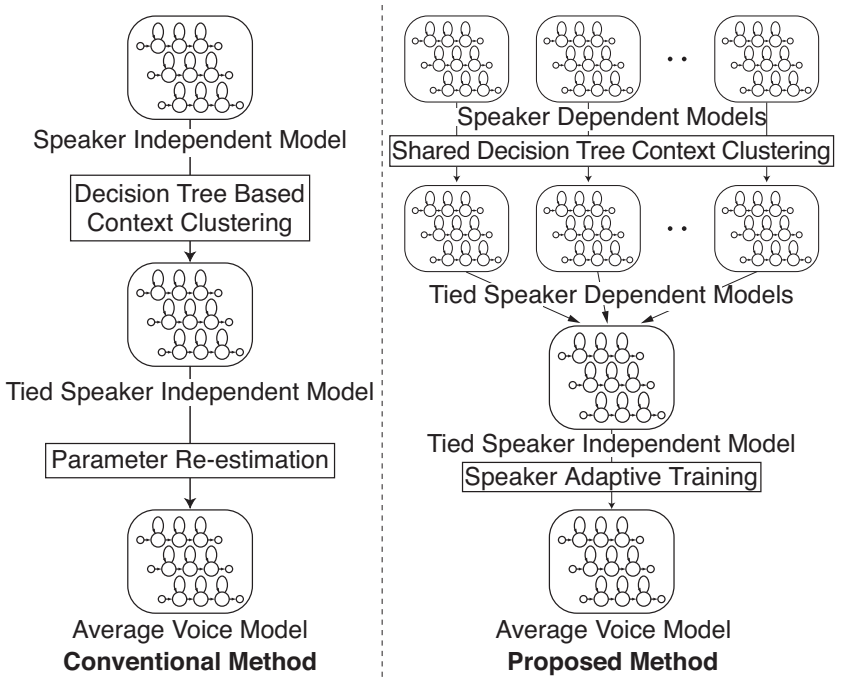


Figure 5.4: Block diagrams of the training stage of the average voice model.

we briefly review the speech synthesis system. The basic structure is the same as the original HMM-based speech synthesis system described in Sect. 3.4 except that the average voice model is used as the set of synthesis units, and adapted to a target speaker using a small amount of speech data uttered by the target speaker. We use the maximum likelihood linear regression (MLLR) algorithm [32] and MSD-MLLR algorithm [7] for spectrum and F0 adaptation, respectively. After the speaker adaptation, speech is synthesized in the same manner as speaker-dependent speech synthesis method. A block diagram of the HMM-based TTS system using speaker adaptation is shown in Fig. 5.3.

5.3.1 Average Voice Model Training

Here we describe a training technique of the average voice model based on both the STC paradigm and the SAT paradigm. A block diagram of the training stage of the average voice model using the proposing technique is

shown on the right side of Fig. 5.4. The left side of Fig. 5.4 shows the training stage using the conventional technique [7]. In the proposing method, first, context dependent models are separately trained for respective speakers. Then, a shared decision tree is constructed using an algorithm described in [36] from the speaker dependent models. All speaker dependent models are clustered using the shared decision tree. A Gaussian pdf of average voice model is obtained by combining all speakers' Gaussian pdfs at every node of the tree. After re-estimation of parameters of the average voice model using speaker adaptive training (SAT) [35] described in Sect. 5.2 with training data of all speakers, state duration distributions are obtained for each speaker. Finally, state duration distributions of the average voice model are obtained by applying the same clustering procedure.

5.4 Experiments

5.4.1 Experimental Conditions

We used the database described in Sect. 4.3 for training HMMs. The database consists of 4 female and 6 male speakers' speech data. We used arbitrarily chosen 3 female and 3 male speakers as training speakers for average voice model, and remaining 1 female and 3 male speakers as target speakers. Speech signals were sampled at a rate of 16kHz and windowed by a 25ms Blackman window with a 5ms shift. The feature vectors consisted of 25 mel-cepstral coefficients including the zeroth coefficient, logarithm of fundamental frequency, and their delta and delta-delta coefficients. The mel-cepstral coefficients were obtained by mel-cepstral analysis [27], [33]. Fundamental frequency values were extracted using an IFAS-based method [40]. The average voice model was trained using 150 sentences for each training speaker. From the results of preliminary experiments, we set the weight for adjusting the number of parameters of the model in STC as $c = 0.4$ and in decision tree based context clustering using MDL criterion [23] as $c = 1$. In SAT procedure, one regression matrix was used for each speaker and was estimated only once. Also we set the number of iterations to 3 for SAT procedure and 5 for EM procedure from the results of preliminary experiments. For comparison,

Table 5.1: The number of distributions after clustering.

	NONE	SAT	STC	STC+SAT
Spectrum	856	856	1251	1251
F0	2742	2742	2217	2217
Duration	1865	1487	2212	1821

we also trained the average voice models with applying STC only and SAT only, respectively. In the case of STC only, EM algorithm is used for the re-estimation process. In the case of SAT only, decision tree based context clustering using MDL criterion, referred to as conventional clustering, is used instead of STC. In the case of the conventional technique [7], neither STC nor SAT is applied.

Table 5.1 shows the number of distributions included in the average voice models after clustering. The entries for "NONE", "SAT", "STC", and "STC+SAT" correspond to the models obtained using the conventional technique [7], SAT only, STC only, and the proposed technique, respectively. In addition, "Spec.", "F0", and "Dur." represent the spectrum, F0, and state duration, respectively.

5.4.2 Subjective Evaluations of Average Voice

We compared the naturalness of the average voice models by a paired comparison test. Subjects were nine persons, and presented a pair of average voices synthesized from different models in random order and then asked which average voice sounded more natural. For each subject, five test sentences were chosen at random from 53 test sentences which were contained in neither training nor adaptation data sentence set.

Figure 5.5 shows the preference scores. It can be seen from the figure that the proposed technique, namely applying both STC and SAT, provides the highest performance. In fact, we have observed that the proposed technique reduces unnaturalness of the average voice speech especially in prosodic features.

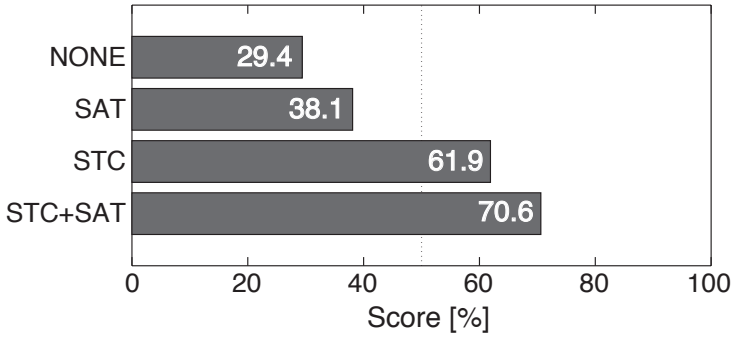


Figure 5.5: Evaluation of naturalness of average voice.

It can also be seen that STC provides the higher performance than SAT. In SAT case, since the conventional clustering is used instead of STC, there existed distributions which had bias depending on speaker and/or gender, and this bias cannot be removed completely even if SAT is applied. On the other hand, in STC case, every distribution reflects the statistics of all speakers. As a result, STC reduced unnaturalness of the average voice speech especially in prosodic features as against SAT.

5.4.3 Objective Evaluations of Synthetic Speech Generated from Adapted Model

We chose a female speaker FTK and three male speakers MMY, MSH, and MTK from the database as the target speakers, who were not included in the training speakers of the average voice model. Based on MLLR-based speaker adaptation technique described in [7], the average voice models were adapted to the target speaker using 10 sentences which were not included in the training data sentence set. In the speaker adaptation using MLLR, one regression matrix was estimated for each stream, and thresholds for traversing regression class tree were set to 1000 for spectrum stream and 100 for F0 stream, respectively. We did not adapt state duration distributions and used the same distributions as the average voice model.

We calculated average mel-cepstral distance and root-mean-square (RMS) error of logarithmic fundamental frequency as the objective evaluations of the synthetic speech generated from the adapted model. 53 test sentences were

Table 5.2: Average mel-cepstral distance in [dB] for 53 test sentences.

Speaker	Model	NONE	SAT	STC	STC+ SAT	SD
MMY	Average	*6.98*	*6.91*	*6.78*	*6.86*	5.15
	Adapted	5.39	5.32	5.32	5.29	
MSH	Average	*6.99*	*7.01*	*6.98*	*7.10*	5.46
	Adapted	5.87	5.89	5.92	5.82	
MTK	Average	*7.55*	*7.65*	*7.51*	*7.61*	5.08
	Adapted	5.93	5.89	5.91	5.81	
FTK	Average	*7.83*	*7.83*	*7.83*	*7.84*	5.45
	Adapted	6.17	6.06	6.19	6.00	

used for evaluation, which were included in neither training nor adaptation data. For the distance calculation, state duration was adjusted after viterbi alignment with the target speaker's real utterance. For comparison, we also evaluated synthesized speech with using speaker dependent units of the target speakers. Each speaker dependent model was trained using 450 sentences uttered by the target speaker, which were not contained in the testing sentence set. The numbers of distributions of the speaker dependent model for MMY were 833, 1410, and 1399 for spectrum, F0, and state duration, respectively, 807, 1276, and 1208 for MSH, 952, 1760, and 1348 for MTK, and 891, 2057, and 1222 for FTK.

Table 5.2 shows the average mel-cepstral distance between spectra generated from each model and obtained by means of analyzing target speaker's real utterance. In the distance calculation, silence and pause regions were eliminated. Table 5.3 shows the RMS logarithmic F0 error between generated logarithmic fundamental frequency and that extracted from target speaker's real utterance. Since F0 value is not observed in the unvoiced region, the RMS logarithmic F0 error is calculated on the region in which both generated contour and real contour are voiced. In these tables, "Average" represents the results for average voice model and "Adapted" represents the results for the model adapted to the target speaker. In addition, "SD" represents the

Table 5.3: RMS logarithmic F0 error in [oct(10^{-1})] for 53 test sentences.

Speaker	Model	NONE	SAT	STC	STC+ SAT	SD
MMY	Average	*1.39*	*1.39*	*1.15*	*1.18*	0.92
	Adapted	1.48	1.41	1.26	1.08	
MSH	Average	*1.80*	*1.80*	*1.70*	*1.59*	0.96
	Adapted	1.42	1.36	1.20	1.06	
MTK	Average	*3.73*	*3.73*	*3.85*	*3.91*	1.16
	Adapted	1.73	1.70	1.44	1.42	
FTK	Average	*3.66*	*3.68*	*3.93*	*4.21*	0.98
	Adapted	1.44	1.48	1.36	1.17	

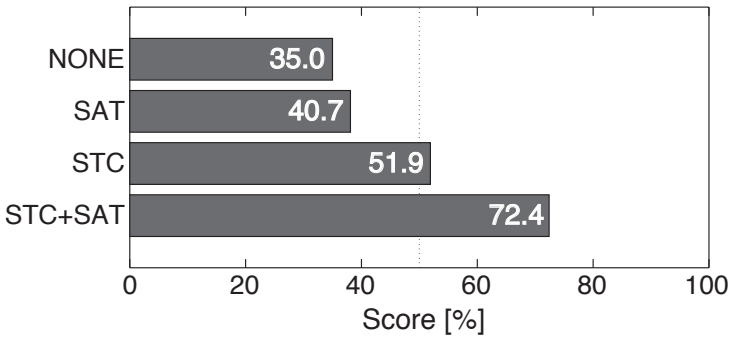
result for speaker dependent model of the target speaker.

From Table 5.2, we can see that mel-cepstrum generated from the adapted model becomes much closer to that generated from the speaker dependent model than the average voice model. And also it can be seen that the mel-cepstrum of the proposed technique becomes slightly closer by comparison with the other training techniques. Table 5.3 shows that F0 value generated from adapted models becomes closer to that generated from speaker dependent models and that the proposed technique gives the least F0 error of the training techniques. Moreover, it is noted that the proposed technique generates F0 whose error is comparable with the speaker dependent model.

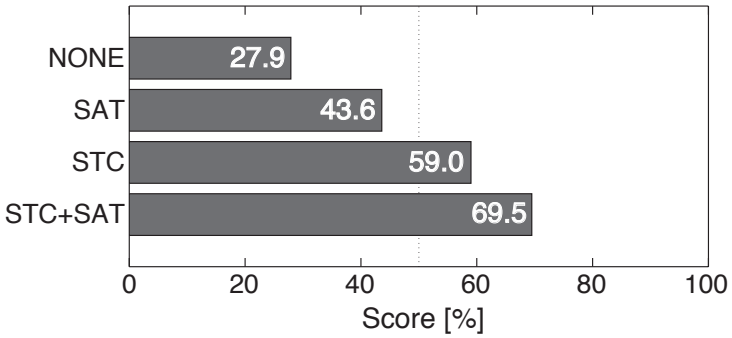
5.4.4 Subjective Evaluations of Synthetic Speech Generated from Adapted Model

We evaluated naturalness of the synthesized speech generated from the models adapted to the target speakers MMY and FTK. Subjects were 7 persons. Other experimental conditions were same as the evaluation test described in Sect. 5.4.2.

Figure 5.6 shows the preference scores. In the figure, (a) is the result for the male target speaker MMY, and (b) is that for the female target speaker



(a) male speaker : MMY

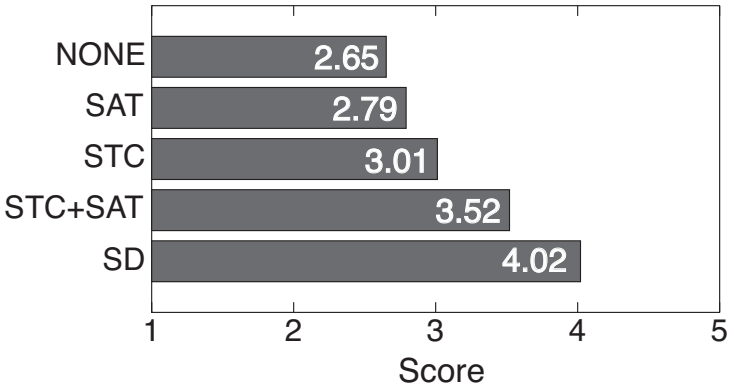


(b) female speaker : FTK

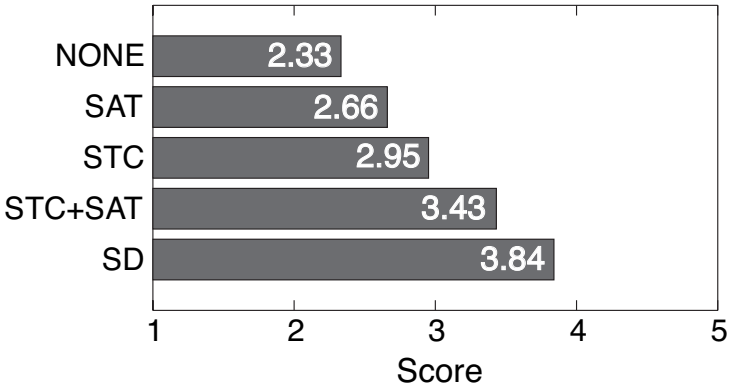
Figure 5.6: Evaluation of naturalness of synthetic speech generated from the adapted model.

FTK. It can be seen that similar results as the average voice were obtained for the synthesized speech from the adapted models. This means that the quality of the average voice crucially affects the quality of synthesized speech from adapted model. Moreover, the proposed technique achieved the best performance of the training methods.

We then conducted a Comparison Category Rating (CCR) test to evaluate voice characteristics and prosodic features of synthesized speech from adapted models. Seven persons listened to 8 sentences of synthesized speech chosen randomly from 53 test sentences and rated their voice characteristics and prosodic features comparing to those of the reference speech. The reference speech was synthesized by a mel-cepstral vocoder. The rating is a 5-point scale, that is, 5 for very similar, 4 for similar, 3 for slightly similar, 2



(a) male speaker : MMY



(b) female speaker : FTK

Figure 5.7: Evaluation of speaker characteristics of synthetic speech generated from the adapted model.

for dissimilar, and 1 for very dissimilar. For comparison, we also evaluated synthesized speech with using speaker dependent units of the target speakers FTK and MMY.

Figure 5.7 shows the results of the CCR test. In the figure, (a) is the result for the target speaker MMY and (b) is that for FTK. The score for "SD" corresponds to the result for synthesized speech using the speaker dependent model of the target speaker. These results confirm again that the proposed technique provides higher performance than the conventional techniques. Moreover, we have observed that the synthetic speech generated from the adapted model using the proposed technique resembles synthesized speech using the speaker dependent model best of all training methods.

In Fig. 5.7, it is seen that the scores for the female speaker FTK are lower than the male speaker MMY. This is due to the fact that extracted spectral envelopes in the mel-cepstral analysis are affected by harmonic structure in a lower frequency region when the fundamental frequency is high. In fact, the fundamental frequency of the female speaker FTK is higher than that of the male speaker MMY. Therefore, if we can use a speech analysis technique which is more robust for the influence of the harmonic structure, scores for the female speaker may become higher.

5.5 Conclusion

We have described a new training method of average voice model for speech synthesis using speaker adaptation. The proposed training method is based on STC and SAT to reduce influence of speaker dependence and improve the quality of the synthetic speech. From the results of subjective tests, we have shown that the proposed training method improves the quality of both average voice and synthetic speech of the given target speaker. Moreover, we have also shown that voice characteristics and prosodic features of synthetic speech generated from the adapted model using the proposed method become closer to the target speaker than the conventional method.

Chapter 6

HSMM-based MLLR & SAT

In speaker adaptation for speech synthesis, it is desirable to convert both voice characteristics and prosodic features such as F0 and phone duration. For achieving the simultaneous adaptation of spectrum, F0 and phone duration in framework of HMM, we need to perform the transformation of not only state output distributions corresponding to spectrum and F0 but also duration distributions corresponding to phone duration. However, it is not straightforward to adapt the state duration because the original HMM does not have explicit duration distributions. Therefore, we utilize a framework of hidden semi-Markov model (HSMM) which is an HMM having explicit state duration distributions and we propose an HSMM-based model adaptation algorithm to simultaneously transform both state output and state duration distributions. Furthermore, we also propose an HSMM-based adaptive training algorithm to normalize both state output and state duration distributions of average voice model at the same time. We incorporate these HSMM-based techniques into our HSMM-based speech synthesis system and show the effectiveness from results of subjective and objective evaluation tests.

6.1 Introduction

In the previous chapter, the HMM-based speaker adaptation and speaker adaptive training were conducted for transforming and normalizing only state output probability distributions corresponding to spectrum and F0 param-

eters of speech data. However, several speakers have characteristic phone duration of speech as well as spectrum and F0. To mimic the speaker characteristic of phone duration, the state duration distributions as well as the output distributions of the average voice model should be simultaneously adapted to the target speaker. Additionally, if the speakers who have characteristic phone duration are used as the training speakers of the average voice model, the speaker characteristics of duration would occur that the state duration distributions have relatively large dependence on speakers and/or gender included in the training speech database. To obtain higher performance in the speaker adaptation, we should normalize the large dependence on speakers and/or gender of the state duration distributions as well as the output distributions.

In this chapter, a speaker adaptation technique and a speaker adaptive training technique are proposed for simultaneously transforming and normalizing spectrum, F0, and duration. The proposed techniques use a framework of hidden semi-Markov model (HSMM) [41]–[43]. The HSMM is an HMM with explicit state duration probability distributions. The framework of HSMM enables us to conduct the simultaneous adaptation of output distributions and state duration distributions and provides a normalization technique of speaker differences and acoustic variability in both output and state duration distributions of the average voice model. We incorporate the HSMM-based techniques into our HSMM-based speech synthesis system and show the effectiveness from results of subjective and objective evaluation tests.

6.2 Hidden Semi-Markov Model

In speaker adaptation for speech synthesis, it is desirable to convert spectrum, F0 and phone duration simultaneously. Hence, in the speaker adaptation using the HMM, we need to perform the transformation of not only state output distributions corresponding to spectrum and F0 but also duration distributions corresponding to phone duration. However, it is not straightforward to adapt the state duration in the HMM framework because the original HMM does not have explicit duration distributions. Therefore, we utilize a frame-

work of hidden semi-Markov model (HSMM) [42] which is an HMM having explicit state duration distributions instead of the transition probabilities to directly model and control phone durations (Figs. 6.1 and 6.2.)[1]. An N-state left-to-right HSMM λ with no skip paths is specified by state output probability distribution $\{b_i(\cdot)\}_{i=1}^N$ and state duration probability distribution $\{p_i(\cdot)\}_{i=1}^N$. In this study, we assume that the i-th state output and duration distributions are Gaussian distributions characterized by mean vector $\boldsymbol{\mu}_i$ and diagonal covariance matrix $\boldsymbol{\Sigma}_i$, and mean m_i and variance σ_i^2, respectively,

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{6.1}$$

$$= \frac{1}{\sqrt{(2\pi)^L |\boldsymbol{\Sigma}_i|}} \exp\left(-\frac{1}{2}(\boldsymbol{o}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}_i^{-1}(\boldsymbol{o}-\boldsymbol{\mu}_i)\right), \tag{6.2}$$

$$p_i(d) = \mathcal{N}(d; m_i, \sigma_i^2) \tag{6.3}$$

$$= \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(d-m_i)^2}{2\sigma_i^2}\right), \tag{6.4}$$

where $\boldsymbol{o}$ is L-dimensional observation vector and d is duration staying in the state i. The observation probability of the training data $\boldsymbol{O} = (\boldsymbol{o}_1, \cdots, \boldsymbol{o}_T)$ of length T, given the model λ can be written as

$$P(\boldsymbol{O}|\lambda) = \sum_{i=1}^{N} \sum_{\substack{j=1 \\ j\neq i}}^{N} \sum_{d=1}^{t} \alpha_{t-d}(j) p_i(d) \prod_{s=t-d+1}^{t} b_i(\boldsymbol{o}_s) \beta_t(i) \tag{6.5}$$

where $^\forall\ t \in [1, T]$, and $\alpha_t(i)$ and $\beta_t(i)$ are the forward and backward probabilities, defined by

$$\alpha_t(i) = \sum_{d=1}^{t} \sum_{\substack{j=1 \\ j\neq i}}^{N} \alpha_{t-d}(j)\, p_i(d) \prod_{s=t-d+1}^{t} b_i(\boldsymbol{o}_s) \tag{6.6}$$

$$\beta_t(i) = \sum_{d=1}^{T-t} \sum_{\substack{j=1 \\ j\neq i}}^{N} p_j(d) \prod_{s=t+1}^{t+d} b_j(\boldsymbol{o}_s)\, \beta_{t+d}(j) \tag{6.7}$$

where $\alpha_0(i) = 1$, and $\beta_T(i) = 1$.

[1]For comparison of the HMM-based and the HSMM-based model adaptation algorithms, refer to [44] [45].

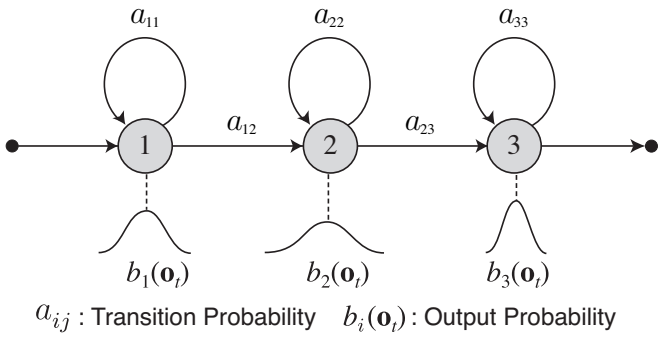


Figure 6.1: Hidden Markov Model

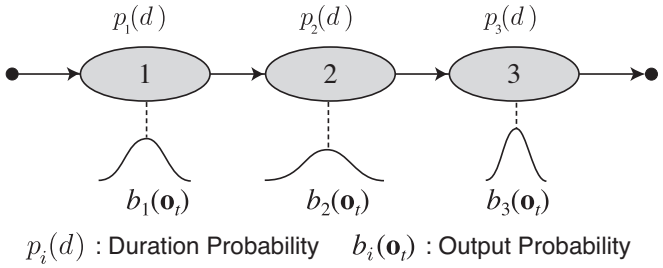


Figure 6.2: Hidden Semi-Markov Model

The conventional speaker independent training of the parameter set λ based on maximum likelihood (ML) criterion can be formulated as follows:

$$\tilde{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, P(\boldsymbol{O}|\lambda). \tag{6.8}$$

Re-estimation formulas based on Baum-Welch algorithm of the parameter set λ are given by

$$\overline{\boldsymbol{\mu}}_i = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\sum_{s=t-d+1}^{t}\boldsymbol{o}_s}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,d} \tag{6.9}$$

$$\overline{\boldsymbol{\Sigma}}_i = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\sum_{s=t-d+1}^{t}(\boldsymbol{o}_s-\overline{\boldsymbol{\mu}}_i)(\boldsymbol{o}_s-\overline{\boldsymbol{\mu}}_i)^\top}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,d} \tag{6.10}$$

$$\overline{m}_i = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,d}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)} \tag{6.11}$$

$$\overline{\sigma}_i^2 = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,(d-\overline{m}_i)^2}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)} \tag{6.12}$$

where $\cdot^\top$ denotes matrix transpose, and $\gamma_t^d(i)$ is a probability generating serial observation sequence $\boldsymbol{o}_{t-d+1}, \cdots, \boldsymbol{o}_t$ at the i-th state and defined by

$$\gamma_t^d(i) = \frac{1}{P(\boldsymbol{O}|\lambda)} \sum_{\substack{j=1 \\ j \neq i}}^{N} \alpha_{t-d}(j)\, p_i(d) \prod_{s=t-d+1}^{t} b_i(\boldsymbol{o}_s)\beta_t(i). \tag{6.13}$$

6.3 HSMM-based MLLR Adaptation

First, we propose an HSMM-based MLLR adaptation [46] to transform both the state output and duration distributions simultaneously. In the HSMM-based MLLR adaptation, mean vectors of state output and duration distributions for the target speaker are obtained by linearly transforming mean vector of state output and duration distributions of the average voice model (Fig. 6.3),

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\zeta}\boldsymbol{\mu}_i + \boldsymbol{\epsilon}, \boldsymbol{\Sigma}_i) \tag{6.14}$$

$$= \mathcal{N}(\boldsymbol{o}; \boldsymbol{W}\boldsymbol{\xi}_i, \boldsymbol{\Sigma}_i) \tag{6.15}$$

$$p_i(d) = \mathcal{N}(d; \chi m_i + \nu, \sigma_i^2) \tag{6.16}$$

$$= \mathcal{N}(d; \boldsymbol{X}\boldsymbol{\phi}_i, \sigma_i^2) \tag{6.17}$$

where $\boldsymbol{\mu}_i$ and m_i are the mean vectors of state output and duration distributions for the average voice model, respectively. $\boldsymbol{W} = [\boldsymbol{\zeta}, \boldsymbol{\epsilon}]$ and $\boldsymbol{X} = [\chi, \nu]$ are $L \times (L+1)$ and 1×2 transformation matrices which transform average voice model into the target speaker for state output and duration distributions, respectively, and $\boldsymbol{\xi}_i = [\boldsymbol{\mu}_i^\top, 1]^\top$ and $\boldsymbol{\phi}_i = [m_i, 1]^\top$ are $(L+1)$-dimensional and 2-dimensional extended mean vectors. $\boldsymbol{\zeta}$ and $\boldsymbol{\epsilon}$ are $L \times L$ matrix and L-dimensional vector, respectively, and both χ and ν are scalar variables.

The HSMM-based MLLR adaptation estimates a set of the transformation matrices $\Lambda = (\boldsymbol{W}, \boldsymbol{X})$ so as to maximize likelihood of adaptation data $\boldsymbol{O}$. The problem of the HSMM-based MLLR adaptation based on ML criterion can be expressed as follows:

$$\tilde{\Lambda} = \left(\tilde{\boldsymbol{W}}, \tilde{\boldsymbol{X}}\right) = \operatorname*{argmax}_{\Lambda} P(\boldsymbol{O}|\lambda, \Lambda) \tag{6.18}$$

where λ is the parameter set of HSMM. Re-estimation formulas based on Baum-Welch algorithm of the transformation matrices Λ can be derived as

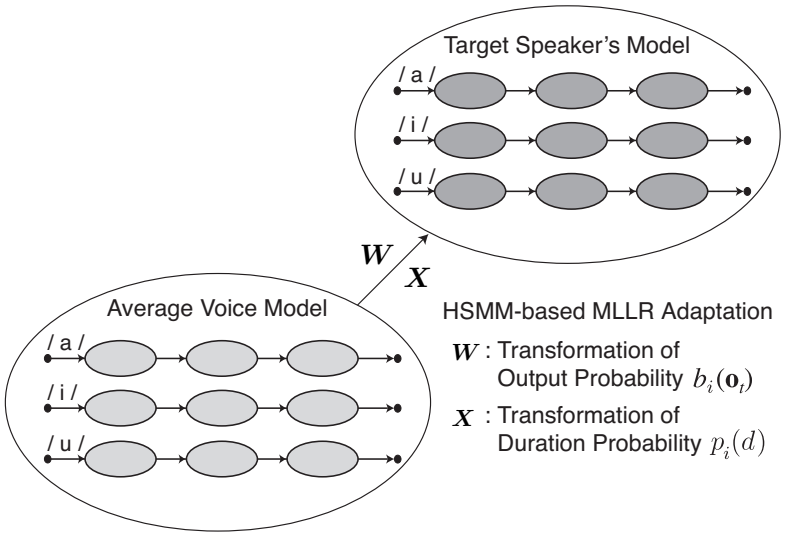


Figure 6.3: HSMM-based MLLR adaptation

follows:

$$\overline{\boldsymbol{w}}_l = \boldsymbol{y}_l \boldsymbol{G}_l^{-1} \tag{6.19}$$

$$\overline{\boldsymbol{X}}_z = \boldsymbol{z} \boldsymbol{K}^{-1} \tag{6.20}$$

where $\boldsymbol{w}_l$ is the l-th row vector of $\boldsymbol{W}$, and $(L+1)$-dimensional vector $\boldsymbol{y}_l$, $(L+1)\times(L+1)$ matrix $\boldsymbol{G}_l$, 2-dimensional vector $\boldsymbol{z}$, and 2×2 matrix $\boldsymbol{K}$ are given by

$$\boldsymbol{y}_l = \sum_{r=1}^{R_b}\sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(r)\ \frac{1}{\Sigma_r(l)} \sum_{s=t-d+1}^{t} o_s(l)\ \boldsymbol{\xi}_r^\top \tag{6.21}$$

$$\boldsymbol{G}_l = \sum_{r=1}^{R_b}\sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(r)\, d\ \frac{1}{\Sigma_r(l)}\, \boldsymbol{\xi}_r\, \boldsymbol{\xi}_r^\top \tag{6.22}$$

$$\boldsymbol{z} = \sum_{r=1}^{R_p}\sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(r)\ \frac{1}{\sigma_r^2}\ d\ \boldsymbol{\phi}_r^\top \tag{6.23}$$

$$\boldsymbol{K} = \sum_{r=1}^{R_p}\sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(r)\ \frac{1}{\sigma_r^2}\ \boldsymbol{\phi}_r\ \boldsymbol{\phi}_r^\top, \tag{6.24}$$

where $\Sigma_r(l)$ is the l-th diagonal element of diagonal covariance matrix $\boldsymbol{\Sigma}_r$,

and $o_s(l)$ is the l-th element of the observation vector $\boldsymbol{o}_s$. Note that $\boldsymbol{W}$ and $\boldsymbol{X}$ are tied across R_b and R_p distributions, respectively.

6.3.1 Implementation Problem

Incorporating the state duration probability greatly increases the computational cost. To keep the computational cost reasonable, we truncated the duration probability $p_r(d)$ at a maximum duration value $\mathcal{D}_r$ within each state. Using this maximum duration value, Eqs. (6.22)–(6.24) can be expressed as follows:

$$\boldsymbol{y}_l = \sum_{r=1}^{R_b} \sum_{t=1}^{T} \sum_{d}^{\min(t,\mathcal{D}_r)} \gamma_t^d(r) \, \frac{1}{\Sigma_r(l)} \sum_{s=t-d+1}^{t} o_s(l) \, \boldsymbol{\xi}_r^\top \tag{6.25}$$

$$\boldsymbol{G}_l = \sum_{r=1}^{R_b} \sum_{t=1}^{T} \sum_{d}^{\min(t,\mathcal{D}_r)} \gamma_t^d(r) \, d \, \frac{1}{\Sigma_r(l)} \boldsymbol{\xi}_r \, \boldsymbol{\xi}_r^\top \tag{6.26}$$

$$\boldsymbol{z} = \sum_{r=1}^{R_p} \sum_{t=1}^{T} \sum_{d}^{\min(t,\mathcal{D}_r)} \gamma_t^d(r) \, \frac{1}{\sigma_r^2} \, d \, \boldsymbol{\phi}_r^\top \tag{6.27}$$

$$\boldsymbol{K} = \sum_{r=1}^{R_p} \sum_{t=1}^{T} \sum_{d}^{\min(t,\mathcal{D}_r)} \gamma_t^d(r) \, \frac{1}{\sigma_r^2} \, \boldsymbol{\phi}_r \, \boldsymbol{\phi}_r^\top . \tag{6.28}$$

To sustain a negligible truncation effect, we have to choose the largest possible $\mathcal{D}_r$ within reasonable computational costs. Hence, we set the maximum duration value $\mathcal{D}_r$ of the duration probability as $\mathcal{D}_r = m_r + k\sigma_r$ and $k = 3$. This enabled the coverage for the duration probability to reach to 99.87%, and results in a practical computational cost.

6.4 HSMM-based SAT Algorithm

Next we propose an HSMM-based speaker adaptive training algorithm [47] for normalizing influence of speaker differences among the training speakers in both state output and duration distributions. The basic idea of the adaptive training algorithm is to view the speaker normalization as a kind of blind problems. In the speaker adaptive training algorithm, the speaker difference between training speaker's voice and canonical average voice is assumed to

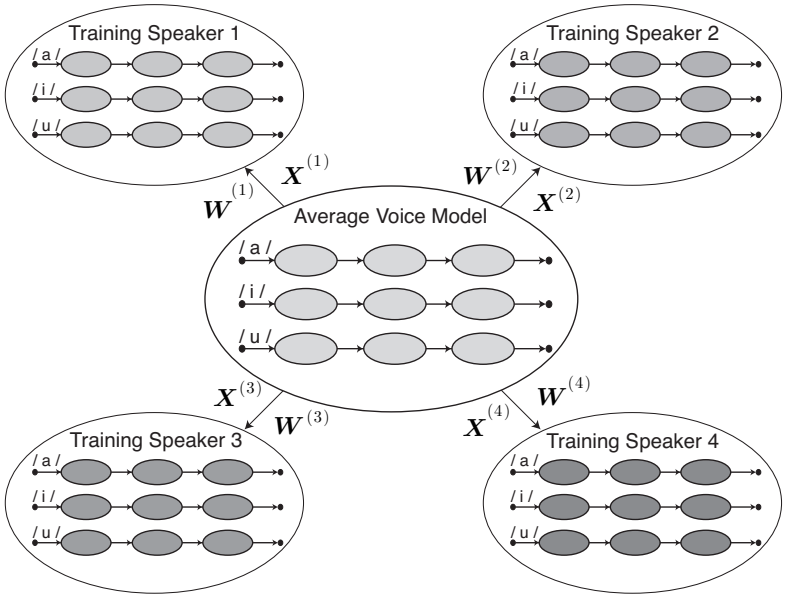


Figure 6.4: HSMM-based speaker adaptive training

be expressed as a simple linear regression function of mean vectors of state output and duration distributions (Fig. 6.4)

$$\boldsymbol{\mu}_i^{(f)} = \boldsymbol{\zeta}^{(f)}\boldsymbol{\mu}_i + \boldsymbol{\epsilon}^{(f)} = \boldsymbol{W}^{(f)}\boldsymbol{\xi}_i \tag{6.29}$$

$$m_i^{(f)} = \chi^{(f)} m_i + \nu^{(f)} = \boldsymbol{X}^{(f)}\boldsymbol{\phi}_i, \tag{6.30}$$

where $\boldsymbol{\mu}_i^{(f)}$ and $m_i^{(f)}$ are the mean vectors of state output and duration distributions for training speaker f, respectively. $\boldsymbol{W}^{(f)} = \left[\boldsymbol{\zeta}^{(f)}, \boldsymbol{\epsilon}^{(f)}\right]$ and $\boldsymbol{X}^{(f)} = \left[\chi^{(f)}, \nu^{(f)}\right]$ are transformation matrices which indicate the speaker difference between training speaker f and average voice in state output and duration distributions, respectively. After the estimating the transformation matrices for each training speaker, a canonical/average voice model is estimated so that the training speaker's model transformed by the matrices maximizes the likelihood for the training data of the training speakers.

Let F be the total number of the training speakers, $\boldsymbol{O} = \{\boldsymbol{O}^{(1)}, \cdots, \boldsymbol{O}^{(F)}\}$ be all training data, and $\boldsymbol{O}^{(f)} = \{\boldsymbol{o}_{1_f}, \cdots, \boldsymbol{o}_{T_f}\}$ be the training data of length T_f for speaker f. The HSMM-based speaker adaptive training simultaneously estimates the parameter set of HSMM λ and a set of the transformation

matrices $\Lambda^{(f)} = (\boldsymbol{W}^{(f)}, \boldsymbol{X}^{(f)})$ for each training speaker so as to maximize likelihood of the training data $\boldsymbol{O}$. The problem of the HSMM-based speaker adaptive training based on ML criterion can be formulated as follows:

$$(\tilde{\lambda}, \tilde{\Lambda}) = \underset{\lambda,\Lambda}{\operatorname{argmax}} P(\boldsymbol{O}|\lambda, \Lambda) = \underset{\lambda,\Lambda}{\operatorname{argmax}} \prod_{f=1}^{F} P(\boldsymbol{O}^{(f)}|\lambda, \Lambda^{(f)}) \tag{6.31}$$

where $\Lambda = (\Lambda^{(1)}, \cdots, \Lambda^{(F)})$ is a set of the transformation matrices for the training speakers. Here we use a three-step iterative procedure to update the parameters [35]. First, we estimate the transformation matrices Λ while keeping λ fixed to the current values. The re-estimation formulas based on Baum-Welch algorithm of the parameter set Λ are identical to Eqs.(6.19)(6.20). We then estimate the mean vectors of λ using the updated transformation matrices while keeping the covariance matrices of λ fixed to the current values. Finally, the covariance matrices of λ are estimated using the updated transformation matrices and the updated mean vectors. The re-estimation formulas of the parameter set λ are given by

$$\begin{aligned}\overline{\boldsymbol{\mu}}_i = &\left[\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i)\, d\ \overline{\boldsymbol{\zeta}}^{(f)\top} \boldsymbol{\Sigma}_i^{-1} \overline{\boldsymbol{\zeta}}^{(f)}\right]^{-1} \cdot \\ &\left[\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i)\, \overline{\boldsymbol{\zeta}}^{(f)\top} \boldsymbol{\Sigma}_i^{-1} \sum_{s=t-d+1}^{t} (\boldsymbol{o}_{s_f} - \overline{\boldsymbol{\epsilon}}^{(f)})\right]\end{aligned} \tag{6.32}$$

$$\overline{\boldsymbol{\Sigma}}_i = \frac{\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i) \sum_{s=t-d+1}^{t} (\boldsymbol{o}_{s_f} - \overline{\boldsymbol{\mu}}_i^{(f)})(\boldsymbol{o}_{s_f} - \overline{\boldsymbol{\mu}}_i^{(f)})^\top}{\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i)\, d} \tag{6.33}$$

$$\overline{m}_i = \frac{\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i)\ \overline{\chi}^{(f)}\ (d - \overline{\nu}^{(f)})}{\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i)\ \overline{\chi}^{(f)2}} \tag{6.34}$$

$$\overline{\sigma}_i^2 = \frac{\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i)\ (d - \overline{m}_i^{(f)})^2}{\sum_{f=1}^{F}\sum_{t=1}^{T_f}\sum_{d=1}^{t} \gamma_t^d(i)}, \tag{6.35}$$

where $\overline{\boldsymbol{\mu}}_i^{(f)} = \overline{\boldsymbol{\zeta}}^{(f)}\overline{\boldsymbol{\mu}}_i + \overline{\boldsymbol{\epsilon}}^{(f)}$ and $\overline{m}_i^{(f)} = \overline{\chi}^{(f)}\overline{m}_i + \overline{\nu}^{(f)}$ are the mean vectors transformed into the training speaker f using the updated mean vectors

and the updated transformation matrices. Here $\overline{\boldsymbol{W}}^{(f)} = \left[\overline{\boldsymbol{\zeta}}^{(f)}, \overline{\boldsymbol{\epsilon}}^{(f)}\right]$ and $\overline{\boldsymbol{X}}^{(f)} = \left[\overline{\chi}^{(f)}, \overline{\nu}^{(f)}\right]$ are the updated transformation matrices for the training speaker f.

6.5 Piecewise Linear Regression

The HSMM-based MLLR adaptation algorithm and the speaker adaptive training can utilize piecewise linear regression functions using multiple transformation matrices in the same manner as HMM-based techniques. For the piecewise linear regression, number and tying topology of the multiple transformation matrices are automatically determined based on tree structure of the distributions and a threshold to specify an expected value of the number of speech samples used for each transformation matrix. Because prosodic feature is characterized by many suprasegmental features, we utilize context decision trees [48] whose questions are related to the suprasegmental features for determining the number and the tying topology of the multiple transformation matrices. The context decision tree is a binary tree and each non-terminal node of the decision tree has a question related to phonetic and linguistic contextual factors, and each terminal node of the decision tree is associated with a distribution in the model. The set of the questions contains many suprasegmental features, such as mora, accentual phrase, part of speech, breath group, and sentence information. Therefore, using a context decision tree for determining the tying topology of the transformation matrices makes it possible to adapt not only frame-based features but also suprasegmental features. Figure 6.5 shows an example of a context decision tree.

6.6 Experiments

6.6.1 Experimental Conditions

To show the effectiveness of simultaneous model adaptation and adaptive training algorithm of spectrum, F0 and duration, we conducted several objec-

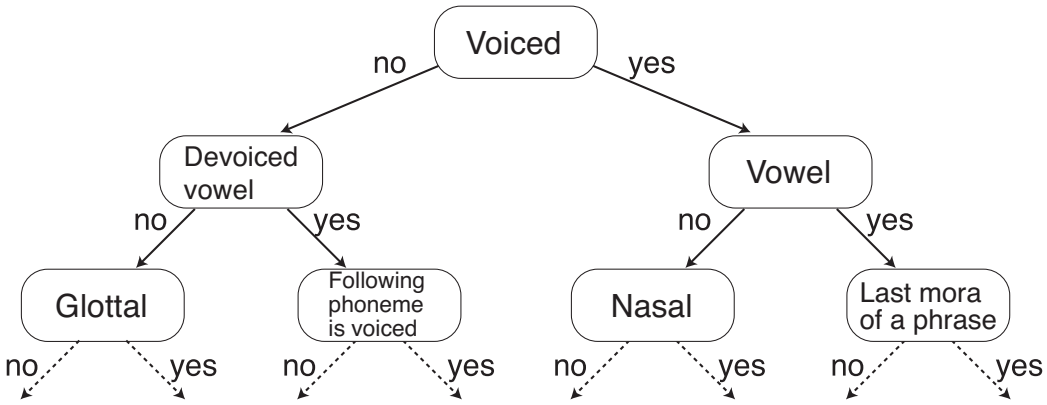


Figure 6.5: An example of the context decision tree.

tive and subjective evaluation tests. We used ATR Japanese speech database (Set B) which contains a set of 503 phonetically balanced sentences uttered by 6 male speakers (MHO MHT MMY MSH MTK MYI) and 4 female speakers (FKN FKS FTK FYM) and a speech database which contains the same sentences as the ATR Japanese speech database uttered by a female speaker (FTY). The average values of logarithm of F0 and mora/sec of each speaker are shown in Figure 6.7. We chose a male speaker MTK and a female speaker FTK as target speakers of the speaker adaptation and used the rest of the speakers as training speakers for the average voice model. In the modeling of synthesis units, we used 42 phonemes, including silence and pause and took the phonetic and linguistic contexts [49] into account.

Speech signals were sampled at a rate of 16kHz and windowed by a 25ms Blackman window with a 5ms shift. The feature vectors consisted of 25 mel-cepstral coefficients [27] [33] including the zeroth coefficient, logarithm of F0 [40], and their delta and delta-delta coefficients. We used 5-state left-to-right context-dependent HSMMs without skip path. The basic structure of the HSMM-based speech synthesis is the same as the HMM-based speech synthesis system [49] except that the HSMMs are used for all stages instead of the HMMs. In the system, gender-dependent average voice models were trained using 450 sentences for each training speaker. The number of training sentences were 2265 sentences and 1812 sentences for male-speaker and female-speaker average voice models, respectively. In the training stage of

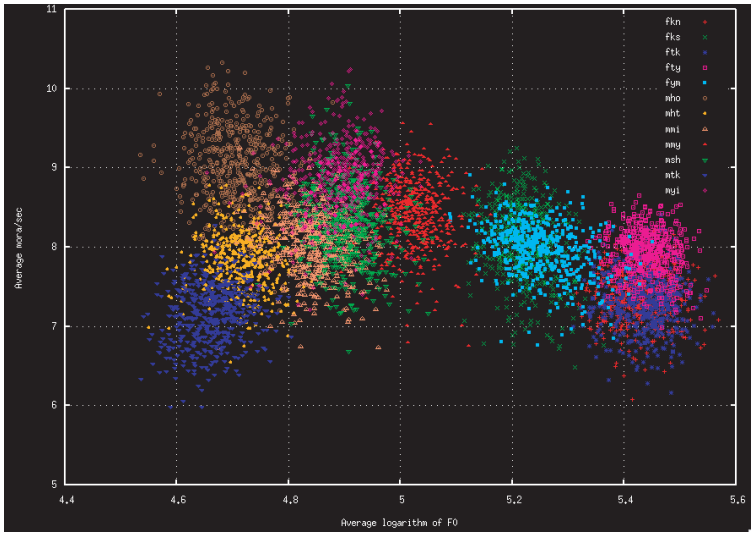


Figure 6.6: Distribution of logF0 and mora/sec of each speaker.

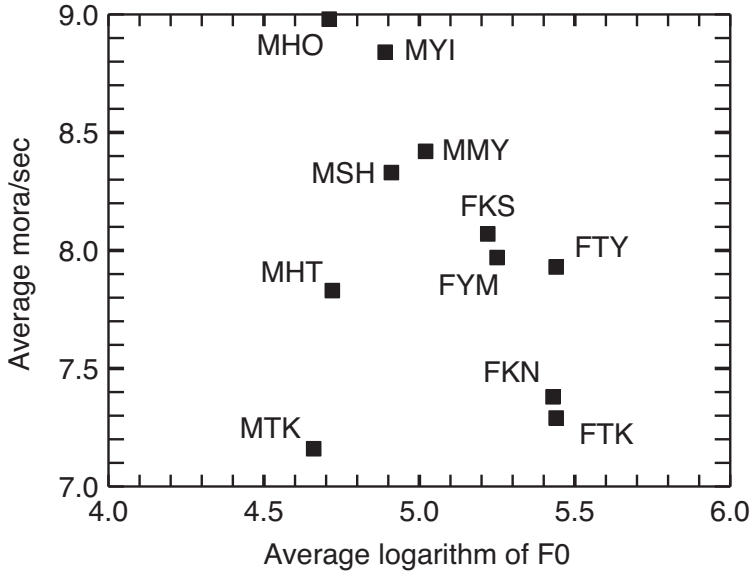


Figure 6.7: Distribution of average logF0 and mora/sec of each speaker.

the average voice models, shared-decision-tree-based context clustering algorithm [49] using minimum description length (MDL) criterion and speaker adaptive training described in Sect. 6.4 were applied to normalize influence of speaker differences among the training speakers. We then adapted the

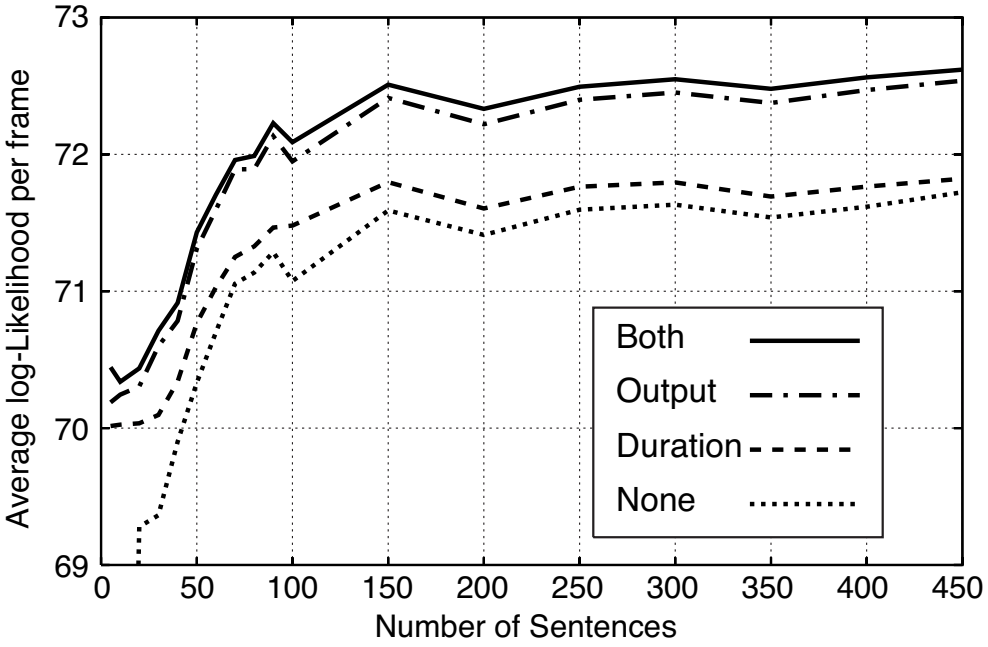


Figure 6.8: Effect of speaker adaptive training of the output and duration distributions. Target speaker is a male speaker MTK.

average voice model to the target speaker using adaptation data whose sentences were included in the training sentences. In the MLLR adaptation and speaker adaptive training, multiple transformation matrices were estimated based on the shared-decision-trees constructed in the training stage of the average voice models. The threshold which specifies an expected value of the number of speech samples used for each transformation matrix was determined based on preliminary objective experimental results. The transformation matrices were diagonal tri-block matrices. The tri-block of the transformation matrices was corresponding to the parts of static, delta, and delta-delta coefficients.

6.6.2 Objective Evaluation of HSMM-based SAT

We firstly evaluated the simultaneous speaker adaptive training for normalizing influence of speaker differences among the training speakers in both state output and duration distributions described in Sect. 6.4. For comparison,

we also trained three kinds of average voice models using the conventional speaker independent training described in Sect. 6.2 for state output and/or duration distributions, respectively. Note that the same topology and the number of distributions based on the shared-decision-trees were used for both the speaker independent training and adaptive training. We then calculated likelihood of the adaptation data for the target speaker as the objective evaluation of the speaker adaptive training. If the speaker differences included in the average voice model are normalized appropriately, the likelihood for the unknown target speaker would increase. The number of the adaptation data is from 5 sentences to 450 sentences.

Figure 6.8 shows the likelihood of the adaptation data for the male speaker MTK. In the figure, "None" represents the results for the average voice model using the conventional speaker independent training for both state output and duration distributions. "Output" and "Duration" represent the results for the average voice models using the speaker adaptive training for only state output or duration distributions, respectively. Then "Both" represents the results for the average voice model using the proposed speaker adaptive training for both state output and duration distributions. From the figure, we can see that the likelihood of the average voice model using the proposed method increases compared to the conventional speaker independent training or the speaker adaptive training for only state output or duration distributions. This is because the HSMM-based speaker adaptive training algorithm can reduce the influence of the speaker differences in both the output and the state duration distribution during re-estimation process, and can suppress inappropriate transformations in the speaker adaptation.

6.6.3 Objective Evaluation of HSMM-based MLLR

We then calculated the average values of logarithm of F0 and mora/sec of the synthetic speech generated from the adapted model using 10 sentences. Fifty test sentences were used for evaluation, which were included in neither training nor adaptation data. The average values of logarithm of F0 and mora/sec of target speakers' speech and the synthetic speech generated from the adapted model are shown in Figure 6.9. For reference, th average values

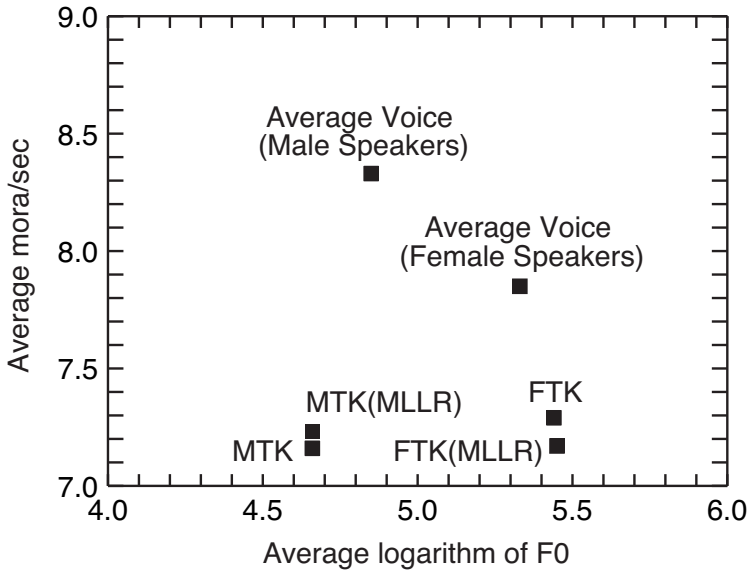


Figure 6.9: Average logF0 and mora/sec of target speakers' speech and synthetic speech generated from the adapted model using 10 sentences.

of logarithm of F0 and mora/sec of average voice, which is synthetic speech generated from the average voice model are also shown in the figure. From the figure, we can see that the average values of logarithm of F0 and mora/sec of the synthetic speech generated from the adapted model are close to those of the target speakers' speech.

Next, we calculated the target speakers' average mel-cepstral distance and root-mean-square (RMS) error of logarithmic F0 and vowel duration as the objective evaluations. The number of the adaptation data is from 5 sentences to 450 sentences. Fifty test sentences were used for evaluation, which were included in neither training nor adaptation data. For the distance calculation of average mel-cepstral distance and RMS error of logarithmic F0, state duration of each model of the HSMM was adjusted after Viterbi alignment with the target speakers' real utterance. For the distance calculation of RMS error of vowel duration, we used manually labeled duration of target speakers' real utterance as the target duration of vowel.

Figures 6.10 and 6.13 show the target speakers' average mel-cepstral distance between spectra generated from the adapted model (MLLR) and ob-

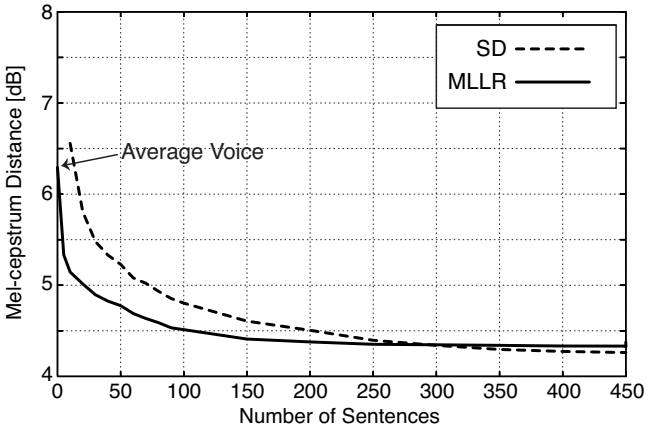


Figure 6.10: Average mel-cepstral distance of male speaker MTK

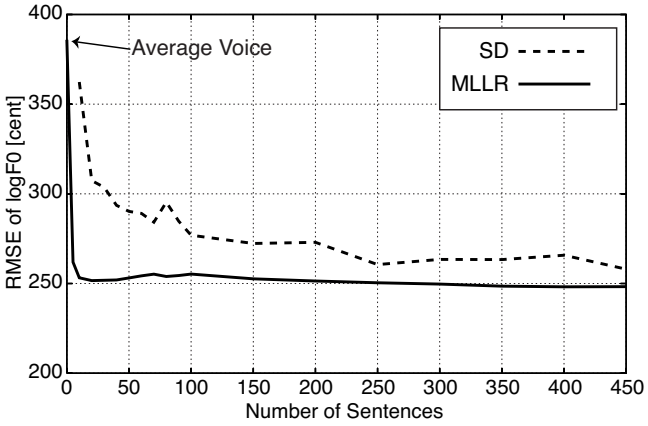


Figure 6.11: RMS logarithmic F_0 error of male speaker MTK

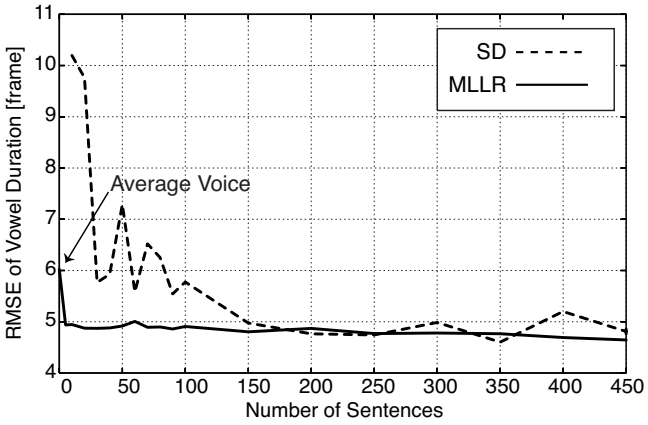


Figure 6.12: RMS error of vowel duration of male speaker MTK

tained by analyzing target speakers' real utterance. For reference, we also show the average distance of spectra generated from speaker dependent (SD) model [50] using the adaptation data as the training data of the SD model.

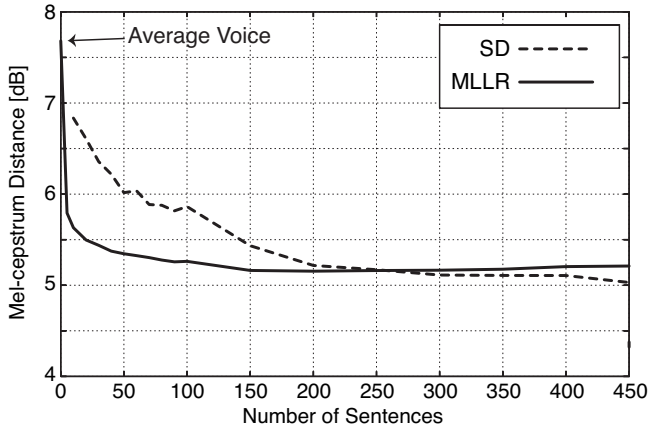


Figure 6.13: Average mel-cepstral distance of female speaker FTK

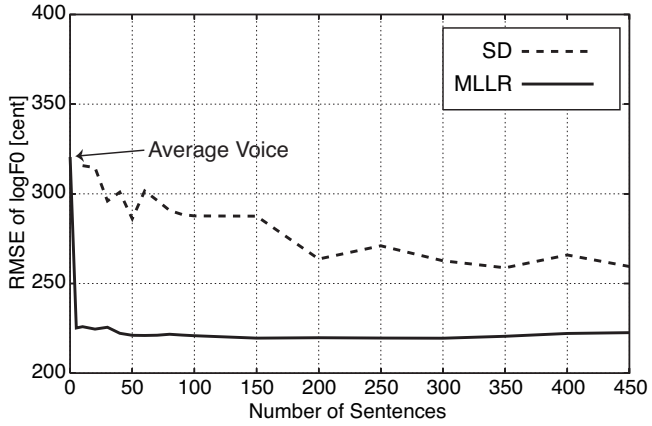


Figure 6.14: RMS logarithmic F_0 error of female speaker FTK

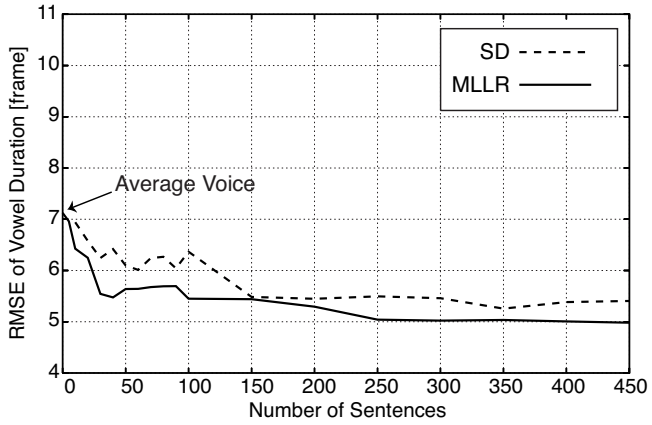


Figure 6.15: RMS error of vowel duration of female speaker FTK

Figures 6.11 and 6.14 show the RMS logarithmic F0 error between F0 patterns of synthetic and real speech. Figures 6.12 and 6.15 show the RMS error between generated duration and that of real utterance. In the distance cal-

culation of mel-cepstral distance, silence and pause regions were eliminated. And since F0 value is not observed in the unvoiced region, the RMS logarithmic F0 error was calculated in the region where both generated F0 and real F0 were voiced.

From these figures, we can see that all features of synthetic speech generated from the adapted model become closer to the target speakers' features than that of average voice, which is synthetic speech generated from the average voice model. Especially when the available data is limited, the adapted model significantly outperforms the speaker dependent model. We can also see that the adapted model gives good results comparable to or a little better than the speaker dependent model even when the sufficient adaptation data is available. Comparing the adaptation of F0 parameters and spectrum parameters, just a few adaptation sentences give good results in the adaptation of the F0 parameter, although about 50 to 150 sentences are needed to obtain good results in the adaptation of the spectral parameters. This is due to the difference of the number of parameters for the features. In this experiments, we used 75-dimensional spectral parameters including the delta and delta-delta parameters, and thus a transformation matrix of the spectral parameters needs a lot of parameters compared to the transformation matrix for F0 parameters which are one-dimensional parameters. As a result, when the available adaptation data is limited, estimation accuracy of the transformation matrix of the spectral parameters decreases compared to that of the transformation matrix for F0 and duration parameters. On the other hand, in the adaptation of duration parameters, the number of adaptation sentences required for the adaptation vary by the target speaker. The fluctuation is thought to be caused by variance of the duration parameters.

6.6.4 Subjective Evaluation

We then conducted a comparison category rating (CCR) test to see the effectiveness of respective transformations of spectral and prosodic features of synthesized speech. We compared the synthesized speech generated from eight models with or without the adaptation of spectrum, F0 and/or duration. The number of the adaptation data was 100 sentences. For reference,

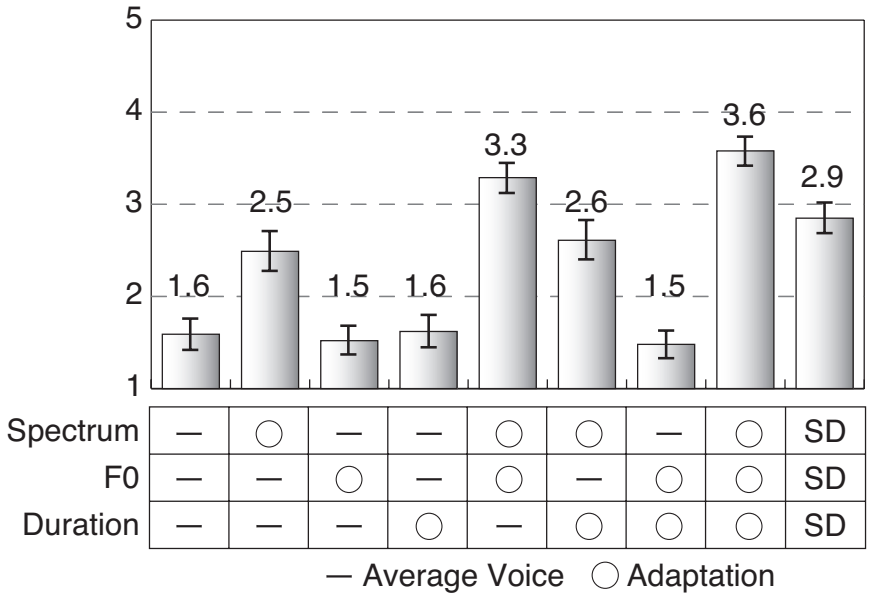


Figure 6.16: Subjective Evaluation of adaptation effects of each feature.

we also compared synthesized speech generated from the SD model using 450 sentences of the target speaker. Eight subjects were first presented reference speech sample and then synthesized speech samples generated from the eight models in random order. The subjects were then asked to rate their voice characteristics and prosodic features comparing to those of the reference speech. The reference speech was synthesized by a mel-cepstral vocoder. The rating was done using a 5-point scale, that is, 5 for very similar, 4 for similar, 3 for slightly similar, 2 for dissimilar, and 1 for very dissimilar. For each subject, five test sentences were randomly chosen from a set of 50 test sentences, which were contained in neither training nor adaptation data.

Figure 6.16 shows the average result of the CCR tests for the male speaker MTK and the female speaker FTK. A confidence interval of 95 % is also shown in the figure. From this figure, we can see that it is not enough for reproducing the speaker characteristics of the target speaker to adapt only voice characteristics or prosodic features, and it is essential to simultaneously adapt both voice characteristics and prosodic features. In the simultaneous adaptation of voice characteristics and prosodic features, this result shows that a combination of spectrum and F0 has a powerful effect on reproduction

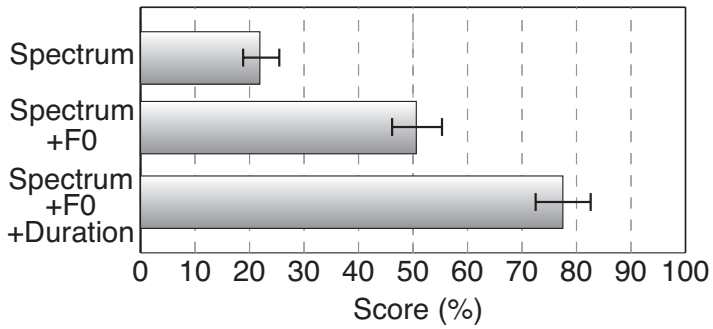


Figure 6.17: Subjective evaluation of simultaneous speaker adaptation.

of the speaker characteristics of the synthesized speech, and furthermore the synthetic speech generated from the model using the simultaneous adaptation of all features have the highest similarity of speaker characteristics of the target speaker. It is interesting to note that the synthesized speech of the proposed method using a small amount of speech data of the target speaker is rated as having higher similarity of speaker characteristics of the target speaker than that of the speaker dependent models. One reason is that F0 parameters generated from the adapted model are more similar to the target speaker than that of the speaker dependent model as can be seen from the results in the Figs. 6.11 and 6.14. Another reason is that the average voice model can utilize a large variety of contextual information included in the several speakers' speech database as a priori information for the speaker adaptation and provide robust basis useful for synthesizing speech of the new target speaker. As a result, synthetic speech having the good similarity of speaker characteristics can be obtained robustly even when speech samples available for the target speaker are very small. In this experiments, the several speakers' speech database for the average voice model have about 3 times as many contextual speech units as the target speaker's speech database. The context-rich speech database and the proposed normalization technique for speaker characteristics would yield a good initial model for the speaker adaptation and the above good results.

We then conducted an ABX comparison test to directly compare the effectiveness of transformations of spectral and prosodic features of synthesized speech. We compared the synthesized speech generated from the adapted

models using adaptation of spectrum, a combination of spectrum and F0, and all features. In the ABX test, A and B were a pair of synthesized speech generated from the adapted models randomly chosen in random order. And X was the reference speech. The reference speech was synthesized by a mel-cepstral vocoder. Eight subjects were presented synthesized speech in the order of A, B, X or B, A, X, and asked to select first or second speech as being similar to X. For each subject, five test sentences were randomly chosen from the same set of test sentences.

Figure 6.17 shows the average preference score of the ABX tests for the male speaker MTK and the female speaker FTK. A confidence interval of 95 % is also shown in the figure. In the figure, "Spectrum","Spectrum+F0", and "Spectrum+F0+Duration" represent the results for the synthetic speech generated from the adapted model using adaptation of spectrum, a combination of spectrum and F0, and all features, respectively. The result confirms again that the synthetic speech generated from the model using the simultaneous adaptation of spectrum, F0, and duration have the highest similarity of speaker characteristics of the target speaker.

6.7 Conclusions

In speaker adaptation for speech synthesis, it is desirable to convert both voice characteristics and prosodic features such as F0 and phone duration. For achieving the simultaneous adaptation of spectrum, F0 and phone duration, we propose the HSMM-based model adaptation algorithm to simultaneously transform both state output and state duration distributions. Furthermore, the HSMM-based adaptive training algorithm is also proposed to normalize both state output and state duration distributions of average voice model at the same time. We have incorporated these HSMM-based techniques into our HSMM-based speech synthesis system and shown the effectiveness of simultaneous model adaptation and adaptive training algorithm of spectrum, F0 and duration from results of subjective and objective evaluation tests.

An issue of the proposed adaptation technique for state duration distribution is that there is a possibility of being adapted to a negative mean

value of the state duration distribution because we assume that state duration distribution is Gaussian distribution. Although we demonstrated that the proposed technique works well and effectively, the state duration distribution is originally defined in the positive area and thus we need to assume a distribution defined in the positive area such as lognormal, gamma, or Poisson distribution for more rigorous modeling and adaptation. Our future work will focus on development of adaptation algorithms of the exponential family of distributions which includes Gaussian, lognormal, gamma, and Poisson distribution using generalized linear regression model [51].

Chapter 7

HSMM-based Speaker Adaptation Algorithms & MAP Modification

In HMM-based speech synthesis, we have to choose the modeling strategy for speech synthesis units depending on the amount of available speech data to generate synthetic speech of better quality. In general, speaker-dependent modeling is an appropriate choice for a large speech data, whereas speaker adaptation with average voice model becomes promising when available speech data of a target speaker is limited. This paper describes several speaker adaptation algorithms and MAP modification to develop consistent method for synthesizing speech in a unified way for arbitrary amount of the speech data. We incorporate the algorithms into our HSMM-based speech synthesis system and show its effectiveness from results of several evaluation tests.

7.1 Introduction

In the HMM-based speech synthesis methods, it is necessary to choose the modeling strategy for speech synthesis unit depending on the amount of available speech data to generate synthetic speech of better quality. In general, speaker-dependent modeling [10] [50] is an appropriate choice for a large

speech data of a target speaker, whereas speaker adaptation with average voice model [49] [47] becomes promising when available speech data of the target speaker is limited. The average voice model can utilize a large variety of contextual information included in the several speakers' speech database as a priori information for the speaker adaptation and provide robust basis useful for synthesizing speech of the new target speaker. As a result, synthetic speech of the target speaker can be obtained robustly even if speech samples available for the target speaker are very small.

For synthesizing speech of good quality for arbitrary amount of the speech data in the speaker adaptation method with average voice model, there are two main problems which we have to deal with. One of them is how to transform effectively the average voice model when the adaptation data for the target speaker is limited. In the previous work [49] [47], the MLLR adaptation was used as the adaptation algorithm of the average voice model. The MLLR adaptation is the most popular and simplest linear regression adaptation, and thus we should explore more effective transformation algorithms to make best use of the limited adaptation data for the target speaker. Another problem is how to estimate the target speaker's model in a unified way when sufficient amount of the adaptation data for the target speaker is available. The speaker adaptation using linear regression has a rough assumption that the target speaker model would be expressed by the linear regression of the average voice model. When the assumption is not inappropriate for a target distribution, the estimation accuracy using the linear regression is consequently less than or comparable to the estimation accuracy using sufficient amount of speech data for the distribution directly. Hence, when sufficient amount of the adaptation data for the target speaker is available, we need to fill a gap between the estimation value using the linear regression and that using speech samples for the target distribution directly.

In this chapter, we explore and compare several speaker adaptation algorithms [32], [52]–[57] to transform more effectively the average voice model into the target speaker's model when the adaptation data for the target speaker is limited. Furthermore, we adopt MAP (Maximum A Posteriori) modification [58] [59] to upgrade the estimation for the distributions having sufficient amount of speech data. When sufficient amount of the adaptation

data is available, the MAP modification theoretically matches the ML estimation. As a result, it is thought that we do not need to choose the modeling strategy depending on the amount of speech data and we would accomplish the consistent method to synthesize speech in the unified way for arbitrary amount of the speech data. We incorporate these algorithms into our speech synthesis system and show its effectiveness from results of subjective and objective evaluation tests.

7.2 HSMM-based Speaker Adaptation

7.2.1 SBR & AMCC

As shown in the preceding chapter, it is desirable to convert spectrum, F0 and phone duration simultaneously in speaker adaptation for speech synthesis. Therefore, in this section, we reformulate the above speaker adaptation algorithms [32], [52]–[59] in the framework of the HSMM.

We firstly describe several simple adaptation algorithms estimating the difference (bias) between the target speaker and the average voice model. In the adaptation algorithms, mean vectors of the state output and duration distributions for the speaker are obtained by adding the bias vector to mean vector of the average voice model,

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}_i + \boldsymbol{\epsilon}, \boldsymbol{\Sigma}_i) \tag{7.1}$$

$$p_i(d) = \mathcal{N}(d; m_i + \nu, \sigma_i^2), \tag{7.2}$$

where $\boldsymbol{\mu}_i$ and m_i are the mean vectors of state output and duration distributions for the average voice model, respectively, and $\boldsymbol{\epsilon}$ and ν are the L-dimensional bias vectors for state output distributions and the scalar bias variable for state duration distributions, respectively. SBR (Signal Bias Removal) [52] [53] estimates a global bias vector for all distributions. In contrast, AMCC (Automatic Model Complexity Control) [54] [55] estimates several bias vectors. Each bias vector is estimated for a cluster of distributions defined by tree structure of distributions (Fig. 7.2). The number of bias vectors is controlled based on heuristic threshold or information criterion such as minimum description length (MDL). Re-estimation formulas based on Baum-

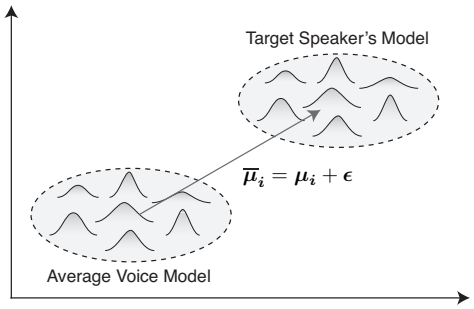


Figure 7.1: Signal Bias Removal

Welch algorithm of the bias vectors for the SBR and the AMCC adaptation can be derived as follows:

$$\overline{\boldsymbol{\epsilon}}_k = \frac{\sum_{r=1}^{R_{kb}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \sum_{s=t-d+1}^{t} (\boldsymbol{o}_s - \boldsymbol{\mu}_r)}{\sum_{r=1}^{R_{kb}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r)\ d} \tag{7.3}$$

$$\overline{\nu}_k = \frac{\sum_{r=1}^{R_{kp}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r)\,(d - m_r)}{\sum_{r=1}^{R_{kp}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r)} \tag{7.4}$$

where R_{kb} and R_{kp} are the number of state output and duration distributions sharing the same bias vector, respectively. In the SBR adaptation, R_{kb} and R_{kp} are the number of all the state output and duration distributions, respectively. In the AMCC adaptation, R_{kb} and R_{kp} are the numbers of the state output and duration distributions belonging to the the cluster k defined by the tree structure, respectively. The distributions belonging to the cluster k share the estimated bias vector $(\overline{\boldsymbol{\epsilon}}_k, \overline{\nu}_k)$.

7.2.2 SMAP

Furthermore, SMAP (Structural Maximum A Posteriori) [56] takes advantage of the tree structure and estimates bias vector for each distribution. In the SMAP adaptation, the bias vector for each node of tree structure is estimated

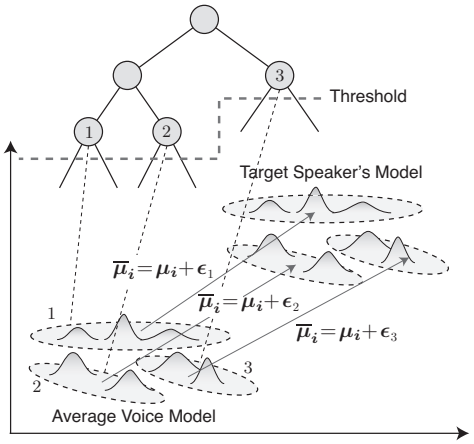


Figure 7.2: Automatic Model Complexity Control

based on maximum a posteriori criterion where a bias vector estimated for a parent node of the current node is used as a parameter of prior distribution. Recursively calculating the MAP estimation from a root node to leaf nodes of the tree structure of distributions, we finally obtain an individual bias vector for each distribution (Fig. 7.3). Re-estimation formulas of the bias vectors for the SMAP adaptation can be derived as follows:

$$\widehat{\boldsymbol{\epsilon}}_k = \frac{\tau_b \, \widehat{\boldsymbol{\epsilon}}_{k-1} + \sum_{r=1}^{R_{kb}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \sum_{s=t-d+1}^{t} (\boldsymbol{o}_s - \boldsymbol{\mu}_r)}{\tau_b + \sum_{r=1}^{R_{kb}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(i) \; d} \tag{7.5}$$

$$\widehat{\nu}_k = \frac{\tau_p \, \widehat{\nu}_{k-1} + \sum_{r=1}^{R_{kp}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \, (d - m_r)}{\tau_p + \sum_{r=1}^{R_{kp}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(i)} \tag{7.6}$$

where k indicates the current node, and $k-1$ indicates the parent node of the node k defined by the tree structure. τ_b and τ_p are positive hyper-parameters of the MAP estimation for the state output and duration distributions, respectively. Note that the estimation at the root node ($k = 1$) of the tree structure is defined by $\widehat{\boldsymbol{\epsilon}}_1 = \overline{\boldsymbol{\epsilon}}_1$ and $\widehat{\nu}_1 = \overline{\nu}_1$.

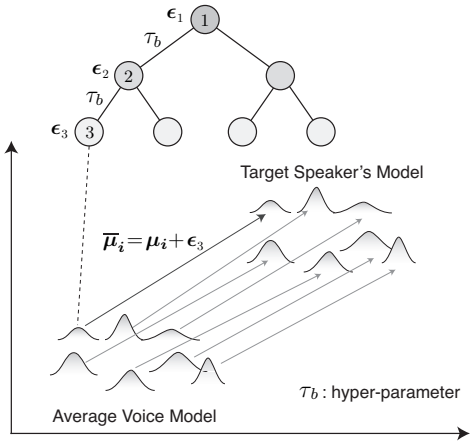


Figure 7.3: Structural Maximum A Posteriori

7.2.3 MLLR

Next, we describe several adaptation algorithms in which linear regression functions are estimated to transform the average voice model into target speaker model. Here we select the following two kinds of linear regression algorithms – MLLR (Maximum Likelihood Linear Regression) [32], and SMAPLR (Structural Maximum A Posteriori Linear Regression) [57].

In MLLR adaptation [32], mean vectors of state output and duration distributions for the target speaker are obtained by linearly transforming mean vector of state output and duration distributions of the average voice model as illustrated in Fig. 7.4,

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\zeta}\boldsymbol{\mu}_i + \boldsymbol{\epsilon}, \boldsymbol{\Sigma}_i) \tag{7.7}$$
$$= \mathcal{N}(\boldsymbol{o}; \boldsymbol{W}\boldsymbol{\xi}_i, \boldsymbol{\Sigma}_i) \tag{7.8}$$
$$p_i(d) = \mathcal{N}(d; \chi m_i + \nu, \sigma_i^2) \tag{7.9}$$
$$= \mathcal{N}(d; \boldsymbol{X}\boldsymbol{\phi}_i, \sigma_i^2) \tag{7.10}$$

where $\boldsymbol{W} = [\boldsymbol{\zeta}, \boldsymbol{\epsilon}] \in \mathcal{R}^{L\times(L+1)}$ and $\boldsymbol{X} = [\chi, \nu] \in \mathcal{R}^{1\times 2}$ are the transformation matrices which transform extended mean vectors $\boldsymbol{\xi}_i = [\boldsymbol{\mu}_i^\top, 1]^\top \in \mathcal{R}^{L+1}$ and $\boldsymbol{\phi}_i = [m_i, 1]^\top \in \mathcal{R}^2$, respectively. $\boldsymbol{\zeta}$ and $\boldsymbol{\epsilon}$ are $L \times L$ matrix and L-dimensional vector, respectively, and both χ and ν are scalar variables. It is straightforward to extend the above global linear regression algorithms to

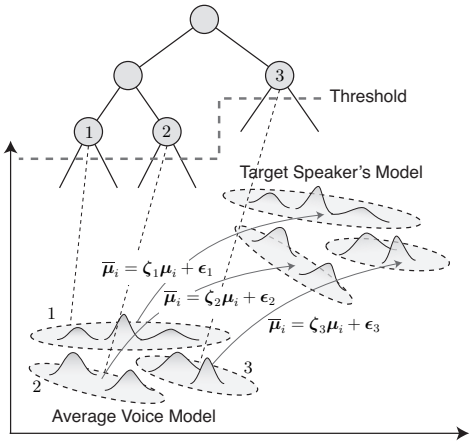


Figure 7.4: Maximum Likelihood Linear Regression

piecewise linear regression algorithms using multiple transformation matrices. Tying topology and the number of the multiple transformation matrices is determined based on tree structure of distributions in the same manner as the AMCC adaptation. We can derive re-estimation formulas based on Baum-Welch algorithm of the transformation matrices $(\boldsymbol{W}_k, \boldsymbol{X}_k)$ for a set of distributions defined by node k of the tree structure as follows:

$$\overline{\boldsymbol{w}}_{lk} = \boldsymbol{y}_{lk} {\boldsymbol{G}_{lk}}^{-1} \tag{7.11}$$

$$\overline{\boldsymbol{X}}_k = \boldsymbol{z}_k {\boldsymbol{K}_k}^{-1} \tag{7.12}$$

where $\boldsymbol{w}_{lk} \in \mathcal{R}^{L+1}$ is the l-th row vector of the $\boldsymbol{W}_k$. In these equations, $\boldsymbol{y}_{lk} \in \mathcal{R}^{L+1}$, $\boldsymbol{G}_{lk} \in \mathcal{R}^{(L+1)\times(L+1)}$, $\boldsymbol{z}_k \in \mathcal{R}^2$, and $\boldsymbol{K}_k \in \mathcal{R}^{2\times 2}$ are given by

$$\boldsymbol{y}_{lk} = \sum_{r=1}^{R_{kb}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \, \frac{1}{\Sigma_r(l)} \sum_{s=t-d+1}^{t} o_s(l) \, \boldsymbol{\xi}_r^\top \tag{7.13}$$

$$\boldsymbol{G}_{lk} = \sum_{r=1}^{R_{kb}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \; d \; \frac{1}{\Sigma_r(l)} \; \boldsymbol{\xi}_r \boldsymbol{\xi}_r^\top \tag{7.14}$$

$$\boldsymbol{z}_k = \sum_{r=1}^{R_{kp}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \; \frac{1}{\sigma_r^2} \; d \; \boldsymbol{\phi}_r^\top \tag{7.15}$$

$$\boldsymbol{K}_k = \sum_{r=1}^{R_{kp}} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \; \frac{1}{\sigma_r^2} \; \boldsymbol{\phi}_r \, \boldsymbol{\phi}_r^\top, \tag{7.16}$$

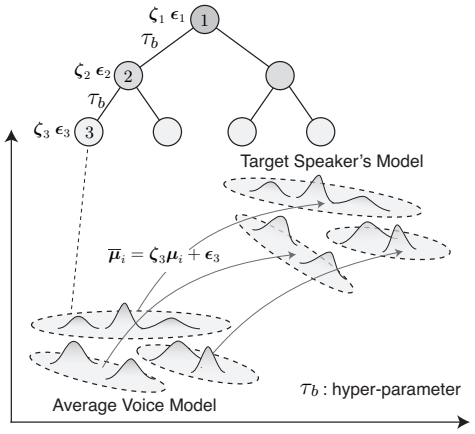


Figure 7.5: Structural maximum a posteriori linear regression

where $\Sigma_r(l)$ is the l-th diagonal element of $\boldsymbol{\Sigma}_r$, and $o_s(l)$ is the l-th element of the observation vector $\boldsymbol{o}_s$. Note that $\boldsymbol{W}_k$ and $\boldsymbol{X}_k$ are tied across R_{kb} and R_{kp} distributions, respectively.

Although the MLLR adaptation needs more parameters for the transformation compared to the SBR or AMCC adaptation, the MLLR adaptation theoretically includes the speaker adaptation using the bias vector and we can expect more appropriate transformation when the available adaptation data is enough for the number of the parameters.

7.2.4 SMAPLR

In SMAPLR adaptation [57], the concept of SMAP adaptation is applied to the ML-based estimation of the multiple transformation matrices for the above piecewise linear regression algorithm, that is, the recursive MAP-based estimation of the transformation matrices from a root node to lower nodes is conducted (Fig. 7.5). As a result, we can make better use of the structural information which the tree strucutre of of distributions has. Re-estimation formulas of the bias vectors for the SMAPLR adaptation can be derived as follows:

$$\widehat{\boldsymbol{w}}_{lk} = (\boldsymbol{y}_{lk} + \tau_b \widehat{\boldsymbol{w}}_{lk-1})(\boldsymbol{G}_{lk} + \tau_b \boldsymbol{I}_{(L+1)\times(L+1)})^{-1} \tag{7.17}$$

$$\widehat{\boldsymbol{X}}_k = (\boldsymbol{z}_k + \tau_p \widehat{\boldsymbol{X}}_{k-1})(\boldsymbol{K}_k + \tau_p \boldsymbol{I}_{2\times 2})^{-1} \tag{7.18}$$

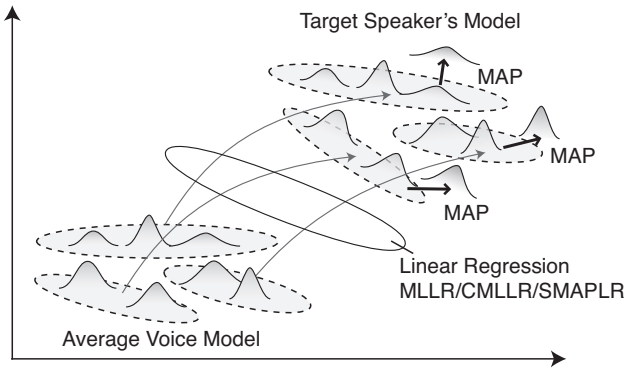


Figure 7.6: Maximum a posteriori modification

where k indicates the current node, $k-1$ indicates the parent node of the node k defined by the tree structure and $\boldsymbol{I}_{y\times y}$ indicates $y \times y$ identity matrix. Note that the prior parameter $\hat{\boldsymbol{w}}_{l0}$ at the root node ($k=1$) is defined by the l-th row vector of the identity transformation matrix $\boldsymbol{W} = [\boldsymbol{I}_{L\times L}, \boldsymbol{0}]$ and $\hat{\boldsymbol{X}}_0$ is defined by the identity transformation matrix $\boldsymbol{X} = [1, 0]$.

7.2.5 MAP Modification

Furthermore, we adopt MAP (Maximum A Posteriori) modification [58] [59]. In the previous speaker adaptation using linear regression, there is a rough assumption that the target speaker model would be expressed by the linear regression of the average voice model. Therefore, by applying the MAP estimation to the model transformed by the linear regression additionally, we can modify and upgrade the estimation for the distribution having sufficient amount of speech samples (Fig. 7.6). The MAP estimations can be estimated as follows:

$$\boldsymbol{\mu}_i^{MAP} = \frac{\tau_b\, \overline{\boldsymbol{\mu}}_i + \sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(i) \sum_{s=t-d+1}^{t} \boldsymbol{o}_s}{\tau_b + \sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(i)\, d} \tag{7.19}$$

$$m_i^{MAP} = \frac{\tau_p\, \overline{m}_i + \sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(i)\, d}{\tau_p + \sum_{t=1}^{T}\sum_{d=1}^{t} \gamma_t^d(i)}, \tag{7.20}$$

where $\overline{\boldsymbol{\mu}}_i$ and $\overline{m}_i$ are the mean vectors transformed by the linear regression, and τ_b and τ_p are positive hyper-parameters of the MAP estimation for the state output and duration distributions, respectively. It is noted that the

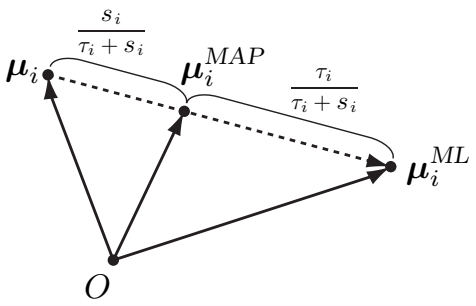


Figure 7.7: Relationship between the MAP and the ML estimates.

MAP estimate values $\boldsymbol{\mu}_i^{MAP}$ and m_i^{MAP} can be viewed as a weighted average of the adapted mean vector $\overline{\boldsymbol{\mu}}_i$ / $\overline{m}_i$ and the ML estimate mean vector $\boldsymbol{\mu}_i^{ML}$ / m_i^{ML}

$$\boldsymbol{\mu}_i^{ML} = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\sum_{s=t-d+1}^{t}\boldsymbol{o}_s}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,d}, \tag{7.21}$$

$$m_i^{ML} = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,\gamma_t^d(i)\,d}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,d}, \tag{7.22}$$

i.e.,

$$\boldsymbol{\mu}_i^{MAP} = \frac{\tau_b}{\tau_b + s_{\text{out}}(i)}\overline{\boldsymbol{\mu}}_i + \frac{s_{\text{out}}(i)}{\tau_b + s_{\text{out}}(i)}\boldsymbol{\mu}_i^{ML} \tag{7.23}$$

$$m_i^{MAP} = \frac{\tau_p}{\tau_p + s_{\text{dur}}(i)}\overline{m}_i + \frac{s_{\text{dur}}(i)}{\tau_p + s_{\text{dur}}(i)}m_i^{ML} \tag{7.24}$$

as shown in Fig. 7.7, where

$$s_{\text{out}}(i) = \sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\,d \tag{7.25}$$

$$s_{\text{dur}}(i) = \sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i). \tag{7.26}$$

When $s_{\text{out}}(i)$ and $s_{\text{dur}}(i)$ equals to zero, i.e., no training sample is available, the MAP estimate is simply the prior mean. On the contrary, when a large number of training samples are used (i.e., $s_{\text{out}}(i) \to \infty$ or $s_{\text{dur}}(i) \to \infty$), the MAP estimate converges to the ML estimate $\boldsymbol{\mu}_i^{ML}$ or m_i^{ML} asymptotically.

Therefore, it is thought that we do not need to choose the modeling strategy depending on the amount of available speech data and we would accomplish the consistent speech synthesis method for synthesizing speech in the unified way for arbitrary amount of the speech data.

7.3 Experiments

7.3.1 Experimental Conditions

To compare the effectiveness of each speaker adaptation algorithm, we conducted several objective and subjective evaluation tests for the synthetic speech using each speaker adaptation algorithm. We used ATR Japanese speech database (Set B) which contains a set of 503 phonetically balanced sentences uttered by 6 male speakers (MHO MHT MMY MSH MTK MYI). We chose a male speaker MTK as a target speaker of the speaker adaptation and used the rest of the speakers as training speakers for the average voice model. In the modeling of synthesis units, we used 42 phonemes, including silence and pause and took the phonetic and linguistic contexts [49] into account.

Speech signals were sampled at a rate of 16kHz and windowed by a 25ms Blackman window with a 5ms shift. The feature vectors consisted of 25 mel-cepstral coefficients [27] [33] including the zeroth coefficient, logarithm of F0 [40], and their delta and delta-delta coefficients. We used 5-state left-to-right HSMMs without skip path. The basic structure of the HSMM-based speech synthesis is the same as the HMM-based speech synthesis system [49] except that the HSMMs are used for all stages instead of the HMMs. In the system, gender-dependent average voice models were trained using 450 sentences for each training speaker. The number of training sentences were 2265 sentences and 1812 sentences for male-speaker and female-speaker average voice models, respectively. In the training stage of the average voice models, shared-decision-tree-based context clustering algorithm [49] using minimum description length (MDL) criterion and speaker adaptive training [47] were applied to normalize influence of speaker differences among the training speakers. We then adapted the average voice model to the target speaker

using adaptation data whose sentences were included in the training sentences. In the all adaptation algorithms except SBR, multiple transformation parameters were estimated. Because prosodic feature is characterized by many suprasegmental features, we utilized context decision trees [48] whose questions were related to the suprasegmental features for determining the number and the tying topology of the multiple transformation parameters. The context decision trees were constructed in the shared-decision-tree-based context clustering of the average voice models. The tuning parameters for each adaptation algorithm, the threshold which specify an expected value of the number of speech samples used for each transformation parameter and hyper-parameters of the MAP estimation, were determined based on preliminary objective experimental results.

7.3.2 Objective Evaluation of Speaker Adaptation Algorithms

Firstly, we calculated the target speakers' average mel-cepstral distance and root-mean-square (RMS) error of logarithmic F0 as the objective evaluations for each speaker adaptation algorithm including MAP modification and speaker dependent (SD) algorithms [50]. By using the shared-decision-tree-based context clustering algorithm for the training speakers of the average voice model and the target speaker of the speaker dependent model at the same time, we constructed the same decision trees common to the training speakers and the target speaker. As a result, we can see the accuracy of the parameter estimation only. The number of the adaptation sentences ranged from three to a hundred. The number of the training sentences for the SD model is 450 sentences. Fifty test sentences were used for evaluation, which were included in neither training nor adaptation data. For the distance calculation, state duration of each model was adjusted after Viterbi alignment with the target speakers' real utterance.

Figures 7.8 and 7.9 show the target speakers' average mel-cepstral distance between spectra generated from each model and obtained by analyzing target speakers' real utterance, and the RMS logarithmic F0 error between F0 patterns of synthetic and real speech. In the distance calculation, silence

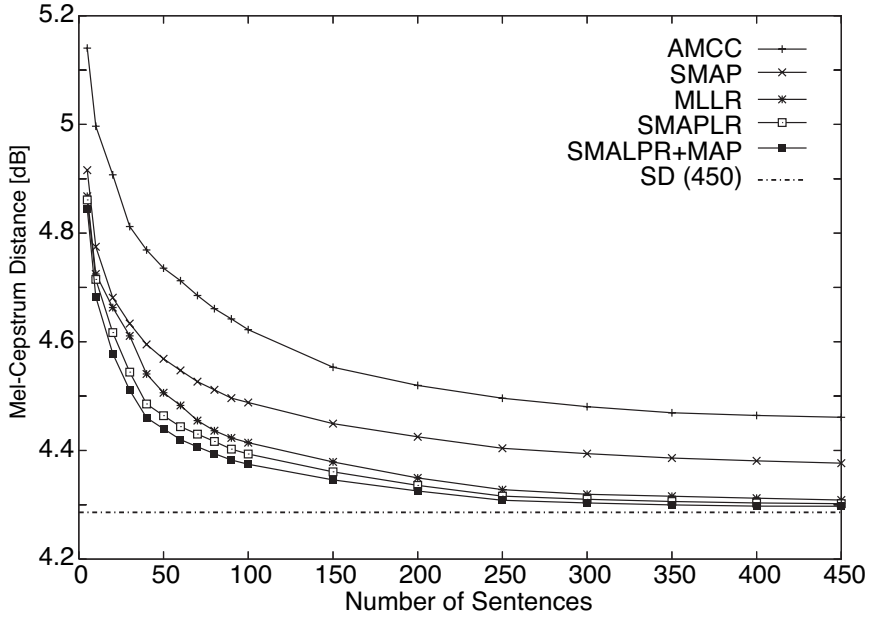


Figure 7.8: Average mel-cepstral distance of male speaker MTK.

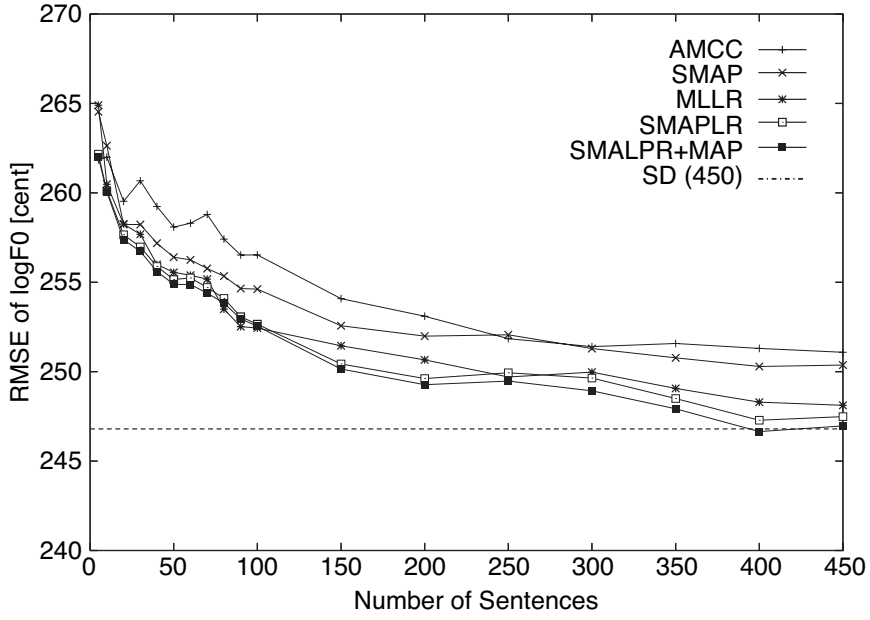


Figure 7.9: RMS logarithmic F_0 error of male speaker MTK.

and pause regions were eliminated. And since F0 value is not observed in the unvoiced region, the RMS logarithmic F0 error was calculated in the region

where both generated F0 and real F0 were voiced. The average mel-cepstral distance and the RMS logarithmic F0 error of synthetic speech generated from the average voice model are 6.3[dB] and 384[cent], respectively. From this figure, it can be seen that making use of the structural information and the suprasegmental information based on the SMAP/SMAPLR adaptation provides synthetic speech having higher similarity to the target speaker than AMCC/MLLR adaptation. And we can see that MAP modification also have a beneficial effect on the improvements even for the case where the adaptation data is small, and furthermore, both feature of synthetic speech generated from the adapted model using the MAP modification converges in the almost same error as that generated from the speaker dependent model when sufficient amount of the adaptation data is available. Note that the convergent mel-cepstral distance and RMS logarithmic F0 error of the adapted model and the speaker dependent model does not match completely because MAP modification for covariances matrices is not conducted here.

7.4 Conclusions

This chapter has described several HSMM-based speaker adaptation algorithms to transform more effectively the average voice model into the target speaker when the adaptation data for the target speaker is limited, and MAP modification algorithm to upgrade the estimation accuracy of the distributions and develop consistent method for synthesizing speech in a unified way for arbitrary amount of the speech data In addition to the above speaker adaptation algorithms, we can reformulate multiple linear regression [60] or Constrained MLLR (CMLLR) [61] [62] and so on. The re-estimation formulas of the adaptation algorithms are described in Appendix D. From the results of subjective and objective evaluation tests, we have evaluated and shown the advantages and effectiveness of the speaker adaptation algorithms and MAP modification.

The target parameters for the speaker adaptation algorithms described above are restricted to the mean vectors of the average voice model. However, we should tune covariance matrices simultaneously to a new speaker because the variance is also one of the important factors affecting speaker

characteristics of synthetic speech. In the CMLLR (Constrained MLLR) adaptation [61] [62], both mean vector and covariance matrix is simultaneously transformed. Therefore, our future work is to integrate the SMAPLR and the CMLLR adaptation.

Chapter 8

Style Modeling

This chapter describes the modeling methods of various emotional expressions and speaking styles in HMM-based speech synthesis. We show two methods for modeling speaking styles and emotional expressions. In the first method called *style-dependent modeling*, each speaking style and emotional expression is modeled individually. In the second one called *style-mixed modeling*, each speaking style and emotional expression is treated as one of contexts as well as phonetic, prosodic, and linguistic features, and all speaking styles and emotional expressions are modeled simultaneously by using a single acoustic model. We chose four styles of read speech — *neutral*, *rough*, *joyful*, and *sad* — and compared the above two modeling methods using these styles. The results of subjective evaluation tests show that both modeling methods have almost the same accuracy, and that it is possible to synthesize speech with the speaking style and emotional expression similar to those of the target speech.

8.1 Introduction

Recent research on speech synthesis has focused on generating emotional expressiveness and various speaking styles in synthesized speech. Although there are many approaches that can be used to add emotional expressiveness and to produce various speaking styles in synthetic speech [63], [64], the emotional expression and speaking style of synthetic speech in most approaches

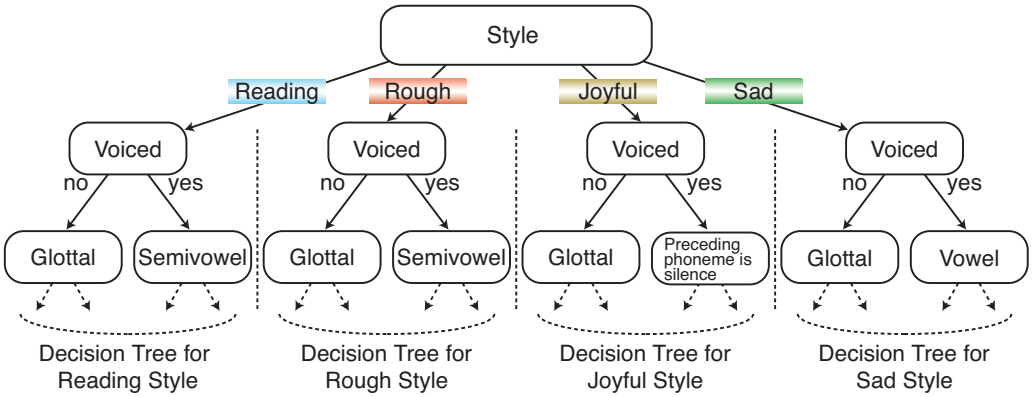
are controlled based on prosodic and other rules such as heuristic adjustments of the F_0 level and range, speech tempo, and loudness. As a consequence, these approaches do not always make it possible to express emotions and speaking styles of all speakers in synthetic speech.

In this chapter, we describe an alternative approach that enables expressing various emotions and/or speaking styles easily and effectively in synthetic speech by using an HMM-based speech synthesis framework. In the proposed approach, speaking styles and emotional expressions are statistically modeled and generated without using heuristic rules to control the prosody and other speech parameters of synthesized speech. We describe two methods for modeling speaking styles and emotional expressions [65]. In the first method, each speaking style and emotional expression is modeled individually. We refer to a set of the resulting models as a style-dependent model and call this method *style-dependent modeling*. In the second method, each speaking style and emotional expression is treated as one of contexts as well as phonetic, prosodic, and linguistic features, and all speaking styles and emotional expressions are modeled simultaneously by using a single acoustic model. We refer to the resulting model as a style-mixed model and call this method *style-mixed modeling*. We compared these two modeling methods using four styles of read speech — *neutral*, *rough*, *joyful*, and *sad*.

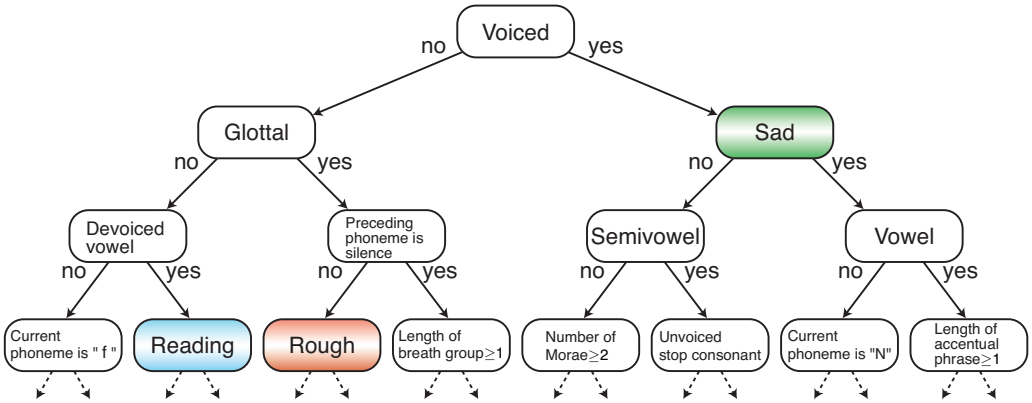
8.2 Stlye Modeling

We developed two acoustic modeling methods — style-dependent modeling and style-mixed modeling — to model various speaking styles in HMM-based speech synthesis. In the following, the term *style* refers to the speaking style including emotional expression.

In the style-dependent modeling method, each style is modeled individually by using an acoustic model. The tree-based context clustering technique described in Sect. 3.3.2 is applied separately to each style's acoustic model as shown in Fig. 8.1 (a). Then a pseudo root node is added to the resulting decision trees of each style to combine the models for all styles into a single acoustic model. One of the advantages of this method is that we can easily add a new style by constructing an acoustic model for it and adding a path



(a) Style-dependent modeling.



(b) Style-mixed modeling.

Figure 8.1: Constructed decision trees in each style modeling.

from the pseudo root node to the root node of the decision tree for the new style.

In the style-mixed modeling method, each style is treated as one of contexts, and the tree-based context clustering technique is applied to all styles at the same time. As a result, all styles are modeled by using a single acoustic model as shown in Fig. 8.1 (b). The styles are automatically split by using style-related questions as well as other contexts during the construction of a decision tree. For the purpose of distinguishing between different styles, we put contextual labels on all phonemes in each sentence. In this method, it is not easy to add new styles because the whole acoustic model must be

Table 8.1: Evaluation of recorded speech samples in four styles.

Speaker	Polite	Rough	Joyful	Sad
MMI	503 (100%)	493 (95%)	499 (98%)	502 (99%)
FTY	503 (100%)	498 (99%)	502 (99%)	502 (99%)

reconstructed. On the other hand, we expect that the sharing of the parameters of similar Gaussian pdfs by several styles would improve the accuracy of these parameters in the Gaussian pdf and would lead to a more compact acoustic model.

8.3 Experiments

8.3.1 Speech Database

To compare the proposed modeling methods, we chose four styles of read speech — *polite*, *rough/impolite*, *joyful*, and *sad* — and constructed speech database [66], which were composed of 503 phonetically balanced sentences obtained from the ATR Japanese speech database. All the sentences were uttered by a male speaker, MMI, and a female speaker, FTY, in all the styles. Both the speakers are professional narrators. We also used speech samples uttered by the same speakers in a *neutral* style for reference purposes.

In the 503 phonetically balanced sentences, there are a number of sentences whose meaning may be unsuitable for several styles except for neutral style. Therefore, we first evaluated whether the recorded speech samples were perceived by listeners as being uttered in the intended styles. Nine male subjects were presented with all 503 sentences uttered in each of the styles and then asked whether they perceived the speech samples as having been uttered in the intended styles.

Table 8.1 shows the number and percentage of sentences which were perceived as having been uttered in the intended style by at least five subjects. It can be seen that almost all of the speech samples in the databases were perceived as having been uttered in the intended styles by a majority of the subjects.

Table 8.2: Classification of styles in the recorded speech.

Speaker	Recorded Speech	Classification (%)					
		Neutral	Polite	Rough	Joyful	Sad	Other
MMI	Neutral	50.7	42.4	3.5	0.0	0.7	2.8
	Polite	38.2	60.4	0.0	1.4	0.0	0.0
	Rough	3.5	2.8	84.0	1.4	2.1	6.2
	Joyful	0.0	0.0	0.0	100	0.0	0.0
	Sad	0.7	6.9	4.2	0.0	79.9	8.3
FTY	Neutral	52.1	43.1	0.7	0.7	3.5	0.0
	Polite	38.9	58.3	0.0	2.1	0.7	0.0
	Rough	0.7	0.0	98.6	0.0	0.7	0.0
	Joyful	1.4	6.9	0.0	91.0	0.0	0.7
	Sad	0.0	0.0	0.0	1.4	98.6	0.0

We then conducted a subjective evaluation test to classify the speech samples of the recorded speech into five groups depending on the style of speech. Nine male subjects were asked to assign eight test sentences chosen at random from 53 test sentences to a *neutral*, *polite*, *rough*, *joyful*, or *sad* group. Speech samples that were not put by the subjects into one of these groups were classified as "other".

Table 8.2 shows the classification results for the recorded speech. These results show that for the rough, joyful, and sad styles, most of the speech samples were perceived by the subjects as having been uttered in the intended styles. However, the speech samples in the neutral and polite styles were perceived as being very similar. We therefore excluded the polite-style speech samples from the following experiments.

8.3.2 Experimental Conditions

We used 42 phonemes including silence and pause and took the phonetic and linguistic contexts described in Sect. 4.3 and a style context into account. The basic structure of HMM-based speech synthesis system used in this study is the same as that of the conventional HMM-based speech synthesis system [10]

except that the labels for the target speaking styles and emotional expressions are given with a target text in the synthesis stage. Both the style-dependent and style-mixed models were trained using 450 sentences for each style. Other experimental coditions are the same as Sect. 4.3.

Table 8.3 shows the number of distributions in each model after decision-tree-based context clustering using the MDL criterion, respectively. The entries for the *style-dependent* and *style-mixed* columns in the table show the number of distributions in the style-dependent and style-mixed models, respectively; the entries for the *neutral*, *rough*, *joyful*, and *sad* columns show the number of distributions for each style in the style-dependent model. The abbreviations *Spec.*, F_0, and *Dur.* refer to the spectrum, F_0, and state duration, respectively. Before the decision-tree-based context clustering, the context-dependent HMMs in the models have the same number of distributions for the spectrum, F_0, and state duration; the style-dependent and style-mixed models also have the same number of distributions. From the table, it can be seen that the number of output and duration distributions in the style-mixed model was smaller than in the style-dependent model. This is because similar model parameters among some styles are shared and the number of redundant distributions decreased in the style-mixed model. Figure 8.1 (a) and (b) show parts of the constructed decision trees for the F_0 part in the second state of the HMMs of the style-dependent and style-mixed models, respectively.

8.3.3 Subjective Evaluations of Styles in Synthesized Speech

We conducted a subjective evaluation test to classify the styles of synthesized speech. For comparison, we also conducted a classification test using the recorded speech. Eleven male subjects were asked to classify eight test sentences chosen at random from 53 test sentences not included in the training data as being *neutral*, *rough*, *joyful*, or *sad* depending on the style of speech[1]. Speech samples that were not assigned by the subjects to one of

[1]Several speech samples used in the test are available at http://www.kbys.ip.titech.ac.jp/research/demo/.

Table 8.3: The number of distributions after tree-based context clustering using the MDL criterion.

Speaker		Style-dependent					Style-mixed
		Neutral	Rough	Joyful	Sad	Total	
MMI	Spec.	891	752	808	926	3377	2796
	F_0	1316	1269	1368	1483	5436	4404
	Dur.	1070	1272	1057	950	4349	3182
	Total	3277	3293	3233	3359	13162	10382
FTY	Spec.	698	635	735	680	2748	2269
	F_0	1464	1545	1343	1249	5601	4598
	Dur.	1033	1407	1531	1105	5076	3801
	Total	3195	3587	3609	3034	13425	10668

these groups were classified as "other".

Tables 8.4 and 8.5 show the classification results for the synthesized and recorded speech for the male speaker, MMI, and female speaker, FTY, respectively. In the tables, (a) shows the results for the style-dependent model, (b) shows the results for the style-mixed model, and (c) shows the results for the recorded speech. It can be seen from the results that both the modeling methods had almost the same reproduction performance, and that we could synthesize speech in styles similar to those of the recorded speech. In these experiments, more than 80% of speech samples generated using both models were judged to be similar to those in the target styles. Note that the subjects, the test speech samples presented to each subject, and the number of styles used were not the same as in the test described in Sect. 8.3.1. As a result, some differences were shown in the classification scores for the recorded speech in table 8.4 (c) compared to those shown in table 8.2 (a).

8.3.4 Subjective Evaluations of Naturalness

We conducted a subjective evaluation test to rate the naturalness of the speech synthesized by using the style-dependent model. Ten subjects listened to eight sentences chosen randomly from 53 test sentences and then they rated

Table 8.4: Subjective evaluation of reproduced styles of MMI.

(a) Style-Dependent Model.

Synthetic Speech	Classification (%)				
	Neutral	Rough	Joyful	Sad	Other
Neutral	98.3	0.6	0.0	0.0	1.1
Rough	6.9	82.3	0.0	0.0	10.8
Joyful	1.1	0.0	94.9	0.0	4.0
Sad	0.6	1.1	0.0	94.9	3.4

(b) Style-Mixed Model.

Synthetic Speech	Classification (%)				
	Neutral	Rough	Joyful	Sad	Other
Neutral	98.9	0.0	0.0	0.0	1.1
Rough	2.8	89.8	0.0	1.1	6.3
Joyful	0.6	0.0	96.0	0.0	3.4
Sad	0.0	0.6	0.0	96.0	3.4

(c) Recorded Speech.

Recorded Speech	Classification (%)				
	Neutral	Rough	Joyful	Sad	Other
Neutral	96.6	2.2	0.6	0.6	0.0
Rough	2.2	96.0	0.6	0.6	0.6
Joyful	1.1	1.1	97.8	0.0	0.0
Sad	0.0	0.6	0.0	99.4	0.0

the naturalness of the synthesized speech. A 3-point scale was used with 3 for "good", 2 for "acceptable", and 1 for "bad".

Figure 8.2 shows the results of the rating test. The scores shown in the figure are the results for the synthesized speech in neutral, rough, joyful and sad styles, respectively. From these results, we can see that this modeling method could generate the synthesized speech with relatively good naturalness in neutral, joyful, and sad styles. However, the scores for the rough-style speech samples are relatively lower than for the samples in the other styles. This is because the phoneme boundaries are unclear in rough-style speech

Table 8.5: Subjective evaluation of reproduced styles of FTY.

(a) Style-Dependent Model.

Synthetic Speech	Classification (%)				
	Neutral	Rough	Joyful	Sad	Other
Neutral	92.5	1.9	5.0	0.0	0.6
Rough	3.1	85.6	1.3	9.4	0.6
Joyful	8.8	0.0	90.6	0.0	0.6
Sad	3.8	6.9	0.0	88.7	0.6

(b) Style-Mixed Model.

Synthetic Speech	Classification (%)				
	Neutral	Rough	Joyful	Sad	Other
Neutral	90.0	1.9	7.5	0.6	0.0
Rough	0.6	90.0	0.0	8.1	1.3
Joyful	3.1	1.9	92.5	0.0	2.5
Sad	1.3	5.6	0.0	91.8	1.3

(c) Recorded Speech.

Recorded Speech	Classification (%)				
	Neutral	Rough	Joyful	Sad	Other
Neutral	97.2	0.0	2.8	0.0	0.0
Rough	0.0	98.9	0.0	1.1	0.0
Joyful	4.0	0.0	96.0	0.0	0.0
Sad	1.7	1.1	0.0	96.6	0.6

samples.

Finally, we compared the naturalness of the synthesized speech generated by the style-dependent and style-mixed models for the male speaker, MMI, by using a paired comparison test. Sixteen male subjects were presented, in random order, with a pair of same-style speech samples synthesized using the two models, and then they were asked which synthesized speech sounded more natural. For each subject, four test sentences were chosen at random from 53 test sentences not included in the training data.

Figure 8.3 shows the preference scores. It can be seen from the figure that

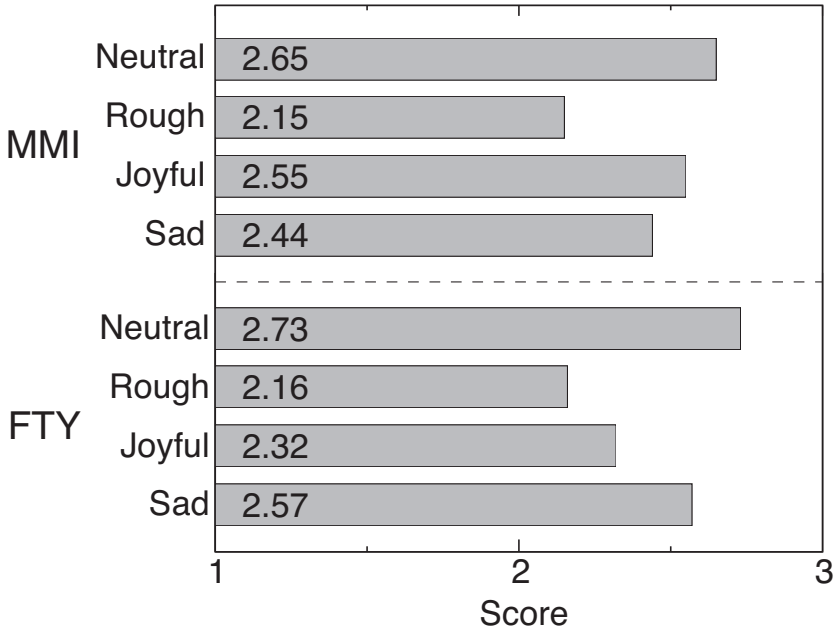


Figure 8.2: Subjective evaluation of naturalness of speech synthesized using style-dependent modeling.

the naturalness of the speech samples synthesized using the two methods was almost the same, although the number of output and duration distributions in the style-mixed model was smaller than in the style-dependent model. From this result, we can conclude that style-mixed modeling is more effective for modeling speech in different styles than the style-dependent modeling.

8.4 Conclusions

We have presented an approach to realizing various speaking styles and emotional expressions in synthetic speech using HMM-based speech synthesis. We have developed two methods for modeling speaking styles and emotional expressions — the style-dependent modeling and style-mixed modeling. We have shown that the two modeling methods have almost the same performance in the subjective evaluation tests, and that it is possible to synthesize speech with speaking styles and emotional expressions similar to those of the recorded speech. In addition, it is also shown that the style-mixed

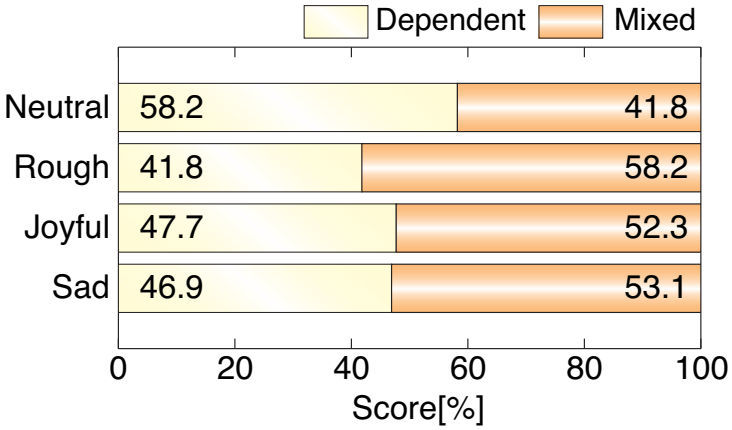


Figure 8.3: Paired comparison test to assess the naturalness of synthesized speech generated using the style-dependent and style-mixed models for MMI.

modeling method can give fewer output and duration distributions than the style-dependent modeling method. Future work will focus on developing variability of speaking styles and emotional expressions using style adaptation techniques.

Chapter 9

Conclusions and Future Work

In this thesis, we have described a novel speech synthesis framework "Average-Voice-based Speech Synthesis." It started as one of the additional ability of the HMM-based speech synthesis, but it finally becomes another speech synthesis framework. By using the speech synthesis framework, synthetic speech of the target speaker can be obtained robustly and steadily even if speech samples available for the target speaker are very small. This speech synthesis framework consists of speaker normalization algorithm for the parameter clustering, speaker normalization algorithm for the parameter estimation, the transformation/adaptation part, and modification part of the rough transformation.

In the decision-tree-based context clustering for average voice model, the nodes of the decision tree do not always have training data of all speakers, and some nodes have data from only one speaker. This speaker-biased node causes degradation of quality of average voice and synthetic speech after speaker adaptation, especially in prosody. In chapter 4, we have proposed a new context clustering technique, named "shared-decision-tree-based context clustering" to overcome this problem. Using this technique, every node of the decision tree always has training data from all speakers included in the training speech database. As a result, we can construct decision tree common to all training speakers and each distribution of the node always reflects the statistics of all speakers.

However, when training data of each training speaker differs widely, the

distributions of the node often have bias depending on speaker and/or gender and this will degrade the quality of synthetic speech. Therefore, in chapter 5, we have incorporated "speaker adaptive training" into the parameter estimation procedure of average voice model to reduce the influence of speaker dependence. In the speaker adaptive training, the speaker difference between training speaker's voice and average voice is assumed to be expressed as a simple linear regression function of mean vector of the distribution and a canonical average voice model is estimated using the assumption.

In speaker adaptation for speech synthesis, it is desirable to convert both voice characteristics and prosodic features such as F0 and phone duration. Therefore, in chapter 6, we utilize a framework of "hidden semi-Markov model" (HSMM) which is an HMM having explicit state duration distributions and we have proposed an HSMM-based model adaptation algorithm to simultaneously transform both state output and state duration distributions. Furthermore, we have also proposed an HSMM-based speaker adaptive training algorithm to normalize both state output and state duration distributions of average voice model at the same time.

In chapter 7, we have explored several speaker adaptation algorithms to transform more effectively the average voice model into the target speaker's model when the adaptation data for the target speaker is limited. Furthermore, we have adopted "MAP (Maximum A Posteriori) modification" to upgrade the estimation for the distributions having sufficient amount of speech data. When sufficient amount of the adaptation data is available, the MAP modification theoretically matches the ML estimation. As a result, it is thought that we do not need to choose the modeling strategy depending on the amount of speech data and we would accomplish the consistent method to synthesize speech in the unified way for arbitrary amount of the speech data.

9.1 Future Work

An issue of the HSMM-based model adaptation technique for state duration distribution is that there is a possibility of being adapted to a negative mean value of the state duration distribution because we assume that state dura-

tion distribution is Gaussian distribution. Although we demonstrated that the proposed technique works well and effectively, the state duration distribution is originally defined in the positive area and thus we need to assume a distribution defined in the positive area such as lognormal, gamma, or Poisson distribution for more rigorous modeling and adaptation. Our future work will focus on development of adaptation algorithms of the exponential family of distributions which includes Gaussian, lognormal, gamma, and Poisson distribution using generalized linear regression model [51].

On the other hand, the target parameters for the speaker adaptation algorithms and speaker adaptive training described above are restricted to the mean vectors of the average voice model. However, we should tune covariance matrices simultaneously to a new speaker because the variance is also one of the important factors affecting speaker characteristics of synthetic speech. In the CMLLR (Constrained MLLR) adaptation [61] [62], both mean vector and covariance matrix is simultaneously transformed. Therefore, our next future work is to integrate the SMAPLR and the CMLLR adaptation in the generalized linear regression model.

Furthermore, we should explore more appropriate time-series-model. Although we demonstrated that the HSMM works well and effectively, the HSMM has an assumption that duration distribution of each state is independent. However, it is obvious that duration distribution of each state is not independent and we need to overcome this problem.

Appendix A

Speaker Adaptation Using Suprasegmental Features

A.1 Introduction

Prosodic feature is characterized by many suprasegmental features, as well as frame-based segmental features. Therefore, it is essential to take account of these features in the speaker adaptation. For adapting the suprasegmental features of prosodic features precisely, we propose a novel tying method for the transformation matrices of the MLLR algorithm to allow the incorporation of both the segmental and suprasegmental speech features into the speaker adaptation. The proposed tying method uses regression class trees with contextual information which will be referred to as "context clustering decision trees."

A.2 Adaptation of Suprasegmental Features

In general, it is not always possible to estimate the MLLR transformation matrices for every distribution, because the amount of adaptation data of a target speaker is small. Therefore, we use tree structures to group the distributions in the model and to tie the transformation matrices in each group. In the tree structure, each node specifies a particular cluster of distributions in the model, and those nodes that have a state occupancy count below a

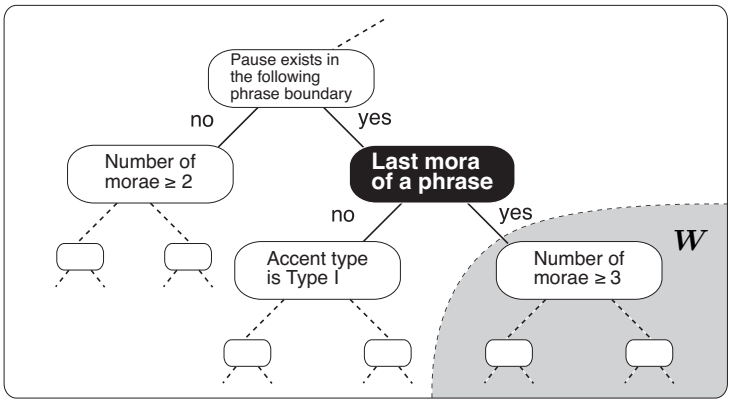


Figure A.1: An example of tying of the regression matrices in the context clustering decision tree.

given threshold are placed in the same regression class as that of their parent node.

To determine the tying topology for the transformation matrices, regression class trees are constructed based on the distance between distributions, such as a Euclidean distance measure [13]. The tying method using the regression class tree has several problems. A key issue is that the regression class tree merely adapts the segmental or frame-based features; in other words, it is difficult to adapt suprasegmental features. This is because the regression class tree is constructed depending on the distance between distributions, and does not reflect the relations among the distributions in the model on the time-axis. However, prosodic features are clearly characterized by many suprasegmental features as well as frame-based features. Therefore, to adapt prosodic features precisely, we have to share the transformation matrices, considering the suprasegmental phonetic and linguistic features. To do this, we utilize context clustering decision trees, constructed during the training stage for tying the transformation matrices instead of a regression class tree.

The context clustering decision tree is a binary tree. Each non-terminal node of the decision tree has a question related to phonetic and linguistic contextual factors, and each leaf/terminal node of the decision tree is associated with a distribution in the model. The set of questions contains many related to suprasegmental features, such as mora, accentual phrase, part of

speech, breath group, and sentence information. Therefore, using a context clustering decision tree for tying the transformation matrices makes it possible to adapt not only frame-based features but also suprasegmental features, if the context clustering decision trees are properly constructed.

Some target speaker have a particular change at the end of phrases. In the context clustering decision tree, this suprasegmental feature was identified by the question "Does the phoneme belong to the last mora of a phrase?" For example, Fig. A.1 shows a part of the context clustering decision tree for the F0 part. In this case, if the transformation matrices below the "yes" node of the question, represented by the shaded area in Fig. A.1, are tied, then the above suprasegmental feature can be reflected in the adapted model.

A.3 Experiments

A.3.1 Experimental Conditions

We used ATR Japanese speech database (Set B) which contains a set of 503 phonetically balanced sentences uttered by 6 male speakers (MHO MHT MMY MSH MTK MYI). The average voice model was trained using 1750 sentences, 350 sentences for each of five male speakers. We set a speaker as the target speaker, and used the rest of the speakers as training speakers for the average voice model. We conducted cross-validation evaluation of above selection.

Speech signals were sampled at a rate of 16kHz and windowed by a 25ms Blackman window with a 5ms shift. The feature vectors consisted of 25 mel-cepstral coefficients including the zeroth coefficient, logarithm of fundamental frequency, and their delta and delta-delta coefficients. We used 5-state left-to-right HMMs. In the training stage of the HMMs, shared-decision-tree-based context clustering algorithm [49] using minimum description length (MDL) criterion and speaker adaptive training [49] were applied to normalize influence of speaker differences among the training speakers. We then adapted the average voice model to the target speaker using 10 or 50 sentences. In the SAT and MLLR algorithm, multiple transformation matrices were estimated for each speaker using shared-decision-trees constructed

Table A.1: The numbers of distributions of the average voice models.

Target Speaker	MHO	MHT	MMY	MSH	MTK	MYI
Spectrum	2991	2924	3027	2877	2750	2946
F0	4462	4333	4701	4663	4507	4531
Duration	3038	2993	3272	3048	2971	3022

Table A.2: The numbers of distributions of the target speaker's dependent models.

Target Speaker	MHO	MHT	MMY	MSH	MTK	MYI
Spectrum	723	962	883	837	977	715
F0	1469	1911	1475	1333	1812	1576
Duration	1075	1093	984	1020	1011	1109

in the training stage. For comparison, we also adapted the average voice model using the conventional regression class tree. Furthermore, we also trained speaker dependent HMMs using 450 sentences for the target speaker as reference models. Tables A.1 and A.2 shows the number of distributions included in the average voice models and the speaker dependent model, respectively. The entries for "Target Speaker" shows the target speakers of the average voice models and the speaker dependent modeling.

A.3.2 Objective Evaluation

Firstly, we calculated the target speakers' average mel-cepstral distance and root-mean-square (RMS) error of logarithmic F0 as the objective evaluations for the use of regression class trees and context clustering decision trees. For reference, we also evaluated average voice generated from the average voice models and the synthetic speech generated from the speaker dependent model. The number of the adaptation sentences is 50 sentences. Fifty test sentences were used for evaluation, which were included in neither training nor adaptation data. For the distance calculation, state duration of each

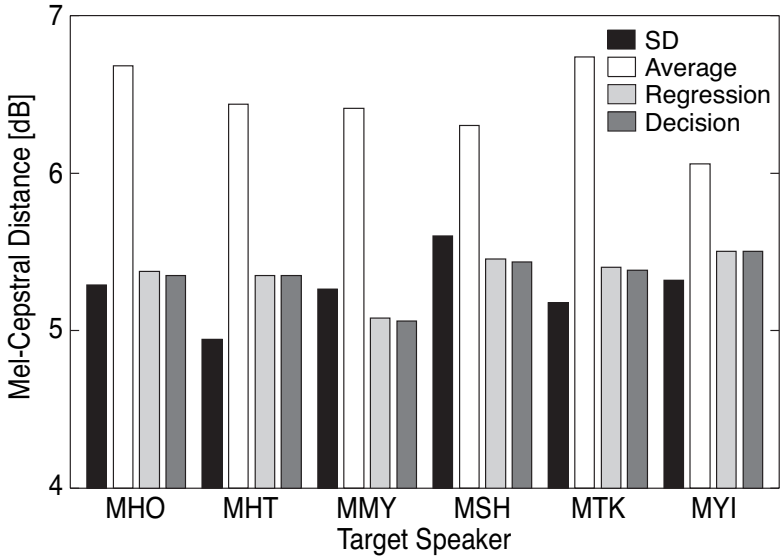


Figure A.2: Average mel-cepstral distance of each target speaker

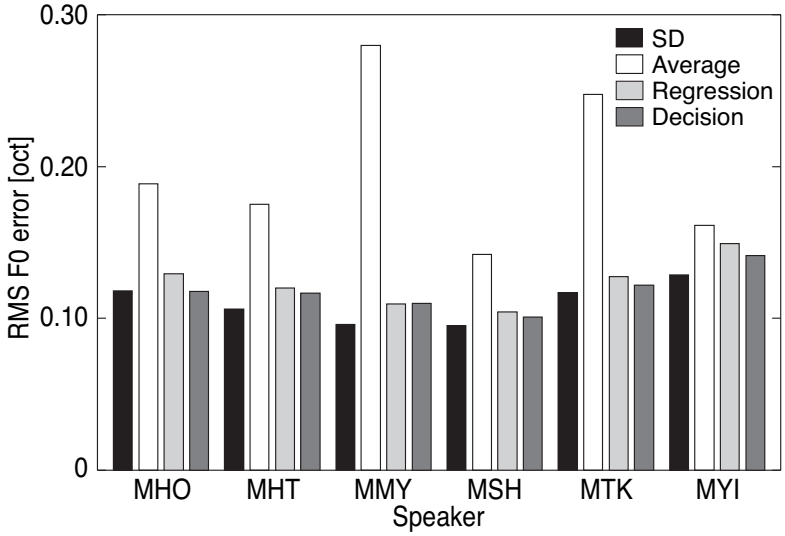


Figure A.3: RMS logarithmic F_0 error of each target speaker.

model was adjusted after Viterbi alignment with the target speakers' real utterance.

Figures A.2 and A.3 show the target speakers' average mel-cepstral distance between spectra generated from each model and obtained by analyzing target speakers' real utterance, and the RMS logarithmic F0 error between

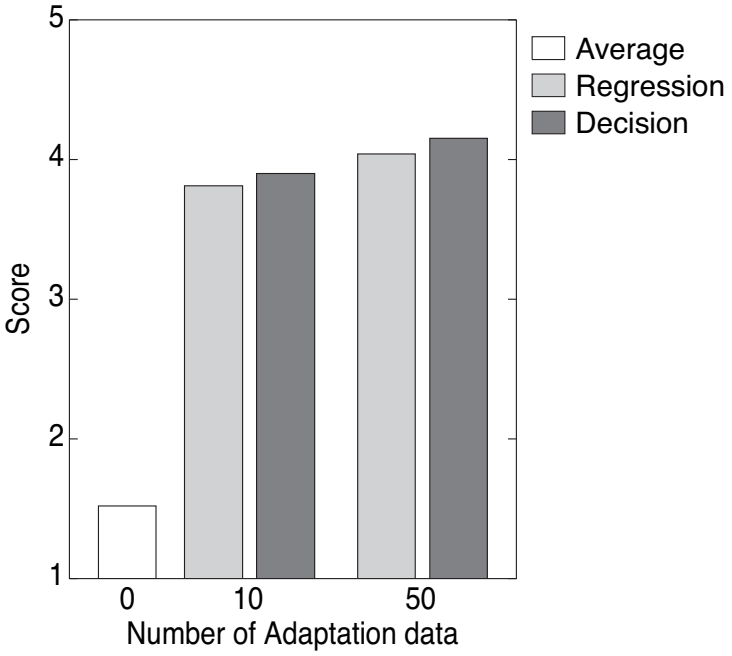


Figure A.4: Result of CCR test for effectiveness evaluation of using the context clustering decision tree.

F0 patterns of synthetic and real speech. In the distance calculation, silence and pause regions were eliminated. And since F0 value is not observed in the unvoiced region, the RMS logarithmic F0 error was calculated in the region where both generated F0 and real F0 were voiced. From this figure, it can be seen that making use of the suprasegmental information of the context clustering decision tree provides synthetic speech having slightly higher similarity to the target speaker in F0 part.

A.3.3 Subjective Evaluation

We conducted a comparison category rating (CCR) test to see the effectiveness of the context clustering decision tree. We compared the synthesized speech generated from the adapted models using regression class trees and context clustering decision trees. The number of the adaptation data was 10 or 50 sentences. For reference, we also compared average voice gener-

ated from the average voice models. 10 subjects were first presented reference speech sample and then synthesized speech samples generated from the adapted models in random order. The subjects were then asked to rate their voice characteristics and prosodic features comparing to those of the reference speech. The reference speech was synthesized speech generated from the speaker dependent model. The rating was done using a 5-point scale, that is, 5 for very similar, 4 for similar, 3 for slightly similar, 2 for dissimilar, and 1 for very dissimilar. For each subject, eight test sentences were randomly chosen from a set of 53 test sentences, which were contained in neither training nor adaptation data.

Figure A.4 shows the average scores of the six target speakers. The result confirms again that the use of context clustering decision trees slightly outperforms that of the regression class trees for determining the tying structure of the MLLR transformation matrix.

A.4 Conclusion

We have proposed a novel tying method for the transformation matrices of the MLLR algorithm to allow the incorporation of both the segmental and suprasegmental speech features into the speaker adaptation. The proposed tying method uses regression class trees with contextual information. From the results of several subjective tests, we have shown that the technique can lead to slight improvements in the adaptation of prosodic features.

Future work will focus on application of the tying method to *style adaptation* [45].

Appendix B

Duration Modeling for HSMM

B.1 Introduction

In chapter 6, we used Gaussian distribution as state duration distribution of HSMM. Although we demonstrated that the proposed technique works well and effectively, the state duration distribution is originally defined in the positive area and thus we need to assume a distribution defined in the positive area such as lognormal, gamma, or Poisson distribution for more rigorous modeling. In this chapter, we show re-estimation formula of the state duration distribution using lognormal, gamma, or Poisson distribution in the HSMM framework.

B.1.1 Lognormal Distribution

We firstly show re-estimation formula of the state duration distribution using lognormal distribution. The i-th state duration distribution using the lognormal distributions is characterized by mean m_i and variance σ_i^2,

$$\log d \sim \mathcal{N}(m_i, \sigma_i). \tag{B.1}$$

The re-estimation formula of the state duration distribution is given by

$$\bar{m}_i = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\log d}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)} \tag{B.2}$$

$$\bar{\sigma}_i = \frac{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)\cdot(\log d - m_i)^2}{\sum_{t=1}^{T}\sum_{d=1}^{t}\gamma_t^d(i)}. \tag{B.3}$$

B.1.2 Gamma Distribution

We next show re-estimation formula of the state duration distribution using Gamma distribution [42]. The i-th state duration distribution using the Gamma distributions is characterized by mean $\frac{\upsilon_i}{\eta_i}$ and variance $\frac{\upsilon_i}{\eta_i^2}$,

$$p_i(d) = \frac{\eta_i^{\upsilon_i}}{\Gamma(\upsilon_i)} d^{\upsilon_i - 1} \exp\left(-\eta_i d\right) \tag{B.4}$$

where $\Gamma(\upsilon_i)$ is defined as

$$\Gamma(\upsilon_i) = \frac{1}{\upsilon_i} \prod_{n=1}^{\infty} (1 + \frac{1}{n})^{\upsilon_i} (1 + \frac{\upsilon_i}{n})^{-1} = \int_0^{\infty} e^{-t} t^{\upsilon_i - 1} dt. \tag{B.5}$$

Then, the re-estimation formula of η_i parameter is given by

$$\bar{\eta}_i = \upsilon_i \cdot \frac{\sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(i)}{\sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(i) d}. \tag{B.6}$$

However, we do not have closed-form of the re-estimation formula of υ_i parameter.

B.1.3 Poisson Distribution

We finally show re-estimation formula of the state duration distribution using Poisson distribution [43]. The i-th state duration distribution using the Poisson distributions is characterized by mean m_i and variance m_i,

$$p_i(d) = \frac{m_i^d e^{-m_i}}{d!} \tag{B.7}$$

The re-estimation formula of m_i parameter is given by

$$\bar{m}_i = \frac{\sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(i)\, d}{\sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(i)}. \tag{B.8}$$

Appendix C

Duration Adaptation

As shown in the chapter 6, appropriate adaptation techniques for both spectral and prosodic features are essential to the speaker adaptation, since the prosodic features, such as fundamental frequency (F0) and duration of a target speaker are frequently much different from the average voice model in general. In [8], the conversion of state duration was based on the statistics from the trellis of HMMs without an explicit state duration probability. Hence, the appropriate statistics for the target speaker cannot be presented when a duration mismatch exists between the average voice and target speaker. In contrast, explicit duration modeling can be incorporated into the HMM-based synthesis framework [50] by introducing a hidden semi-Markov model (HSMM) [42], which is an HMM with explicit state duration probability distributions. By using the statistics from the HSMMs with explicit state duration probability distributions, we can achieve a mathematically rigorous and robust adaptation of state duration as shown in the chapter 6. In this chapter, we compare and show the difference of the HSMM-based adaptation algorithm and the conventional HMM-based adaptation algorithm [8].

C.1 Discussion

In [8] [10], state duration probabilities were estimated approximately on the trellis which was obtained in the embedded training of HMM without explicit duration duration probability for the simplification of implementation.

Table C.1: Comparison of the MLLR adaptation techniques.

	Target pdf	
Method	Output pdf	Duration pdf
Conventional [7]	HMM-based MLLR	–
Conventional [8]	HMM-based MLLR	Approximation of HSMM-based MLLR using HMM's statistics
Proposed	HSMM-based MLLR	HSMM-based MLLR

Moreover, the conversion method of phoneme/state duration [8] was based on the statistics from the trellis of HMMs without explicit state duration probability. As a result, in the conventional method [8], the adapted regression vector for state duration distributions was not re-estimated in a strict way. Therefore it is not clear whether the objective function converges to a critical point. On the other hand, it is straightforward to prove that the objective function $P(\boldsymbol{O}|\lambda, \boldsymbol{W}_i, \boldsymbol{X}_i)$ converges to a critical points using the proposed re-estimation formulae. Furthermore, the proposed adaptation technique for output distributions includes the HMM-based MLLR adaptation of [32] in case of $\mathcal{D} = 1$ and $p_i(1) = 1$. Table C.1 shows the difference of the conventional and proposed MLLR adaptation algorithms.

C.2 Experiments

C.2.1 Speech Database and Experimental Conditions

We used ATR Japanese speech database (Set B) which contains a set of 503 phonetically balanced sentences uttered by 6 male speakers (MHO MHT MMY MSH MTK MYI). We chose male speakers MHO and MHT as target speakers of the speaker adaptation and used the rest of the speakers as training speakers for the average voice model. In the modeling of synthesis units, we used 42 phonemes, including silence and pause and took the phonetic and linguistic contexts [49] into account. Speech signals were sampled at a rate of

16kHz and windowed using a 25ms Blackman window with a 5ms shift. Then mel-cepstral coefficients were obtained using mel-cepstral analysis [33]. The feature vectors consisted of 25 mel-cepstral coefficients including the zeroth coefficient, logarithm of F0, and their delta and delta-delta coefficients.

We used 5-state left-to-right HMMs/HSMMs. We used an HMM-based TTS system [10] and an HSMM-based TTS system [50]. The number of training sentences were 1500 sentences. In the training stage of the average voice models, Decision-tree-based context clustering was applied using the minimum description length (MDL) criterion [23]. We then adapted it to that of one target speaker. The adaptation data included a 50-sentence set (around 3 minutes) of the target speaker taken from training sentences for the speaker-dependent model. The ATR database sentences consist of ten subsets — subsets A, B, ..., and J, and we chose subset A as the adaptation data. The thresholds of the state occupancy count to share the regression matrices for the regression class trees or the context clustering decision trees were set to 1000 for the spectral part, 200 for the F0 part, and 200 for the state duration distributions, respectively. These values were determined based on the preliminary experimental results of informal listening tests for the HMM-based MLLR using several values after [8], and the same values were used for the HSMM-based MLLR. Therefore, they were not necessarily optimal values for the HSMM-based MLLR. Diagonal block transformation matrices were used for the MLLR adaptation. For comparison, we also trained speaker dependent HMMs/HSMMs using 450 sentences for the target speaker as reference models.

C.2.2 Comparison of HMM-based MLLR and HSMM-based MLLR

We compared the naturalness of the synthesized speech for the speech generated from the adapted models using the HMM-based MLLR and the HSMM-based MLLR. 11 subjects were presented a synthesized speech pair randomly generated from the adapted models using HMM-based MLLR and HSMM-based MLLR and asked which speech sounded more natural. For each subject, 15 test sentences were randomly chosen from 53 test sentences.

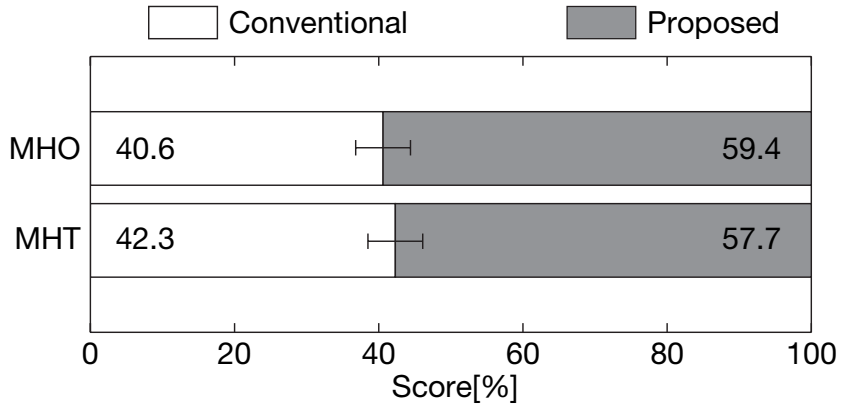


Figure C.1: Preference scores of naturalness of synthesized speech using HMM-based MLLR and HSMM-based MLLR.

Figure C.1 shows the preference scores. We can see that the naturalness of the speech samples from the models adapted using HSMM-based MLLR were significantly better at a 95% confidence level than those adapted using HMM-based MLLR, in both target speakers. In fact, we observed that the synthesized speech using HMM-based MLLR sometimes produced an unnatural prosody in duration.

We then conducted a comparison category rating (CCR) test to see the effectiveness of the HSMM-based MLLR adaptation. We compared the synthesized speech generated from the adapted models using HMM-based MLLR adaptation and HSMM-based MLLR adaptation. For reference, we also compared average voice generated from the average voice models and synthesized speech generated from the speaker dependent model using HMM and HSMM. The number of the adaptation data was 10 or 50 sentences. 17 subjects were first presented reference speech sample and then synthesized speech samples generated from the models in random order. The subjects were then asked to rate their voice characteristics and prosodic features comparing to those of the reference speech. The reference speech was synthesized by a mel-cepstral vocoder. The rating was done using a 5-point scale, that is, 5 for very similar, 4 for similar, 3 for slightly similar, 2 for dissimilar, and 1 for very dissimilar. For each subject, 8 test sentences were randomly chosen from a set of 53 test sentences, which were contained in neither training nor adaptation data.

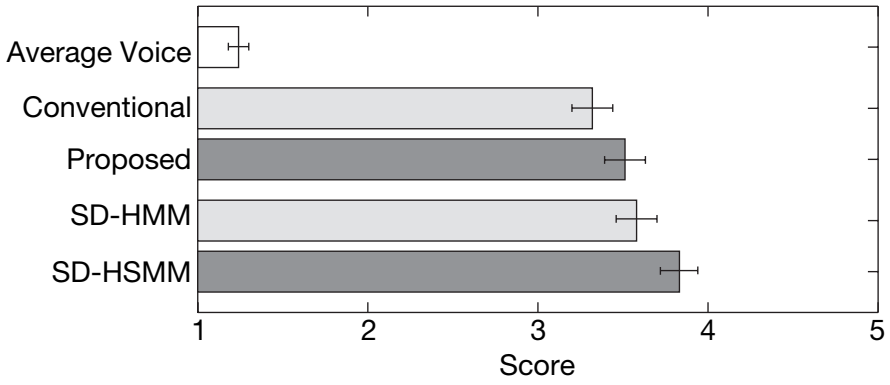


Figure C.2: Result of CCR test for effectiveness evaluation.

Figure C.2 shows the average scores for the target speaker MHT. The result confirms again that

C.3 Conclusion

We have investigated the difference of the HSMM-based adaptation algorithm and the conventional HMM-based adaptation algorithm. From the results of several subjective tests, we have shown that the HSMM-based technique outperforms the conventional HMM-based adaptation algorithm.

Appendix D

Comparative Study of Speaker Adaptation Algorithms

In chapter 7, we described several HSMM-based speaker adaptation algorithms. In addition to the above speaker adaptation algorithms, we can reformulate multiple linear regression [60] or Constrained MLLR (CMLLR) [61] [62] in the framework of the HSMM. In this chapter, we will show some results of the comparative study of the adaptation algorithms.

D.1 Multiple Linear Regression

In the MLLR adaptation, mean vectors for the target speakers are estimated by a simple linear regression using a single average voice model. We can extend the simple linear regression to multiple linear regression using several average voice models,

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \textstyle\sum_{f=1}^{F} \boldsymbol{\zeta}^{(f)} \boldsymbol{\mu}_i^{(f)} + \boldsymbol{\epsilon}, \boldsymbol{\Sigma}_i) \tag{D.1}$$

$$p_i(d) = \mathcal{N}(d; \textstyle\sum_{f=1}^{F} \chi^{(f)} m_i^{(f)} + \nu, \sigma_i^2) \tag{D.2}$$

where F is the number of average voice models and $\boldsymbol{\mu}_i^{(f)}$ and $m_i^{(f)}$ are the mean vectors of the f-th average voice model. Re-estimation formulas based on Baum-Welch algorithm of transformation matrices can be derived in the same manner as the MLLR adaptation.

This algorithm, called ESAT [60], automatically selects or blends several

typical average voice models depending on speaker characteristics of the target speaker. As a result, the ESAT would widely expand the range of the target speaker of the speaker adaptation compared to a single average voice model [67]. However, the ESAT adaptation needs F times as many parameters as the MLLR adaptation needs. Hence, if the adaptation data is not enough for the number of parameters, the accuracy of transformation matrices decreases.

D.2 Constrained MLLR

The target parameters for the speaker adaptations described in chapter 7 are restricted to the mean vectors of the average voice model. However, we should tune covariance matrices simultaneously to a new speaker because the variance is also one of the important factors affecting speaker characteristics of synthetic speech. In the CMLLR adaptation, mean vectors and covariance matrices of state output and duration distributions for the target speaker are obtained by transforming the parameters simultaneously (Fig. D.1) as follows:

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\zeta}'\boldsymbol{\mu}_i - \boldsymbol{\epsilon}', \boldsymbol{\zeta}'\boldsymbol{\Sigma}_i{\boldsymbol{\zeta}'}^\top) \tag{D.3}$$

$$p_i(d) = \mathcal{N}(d; \chi' m_i - \nu', \chi'\sigma_i^2\chi'). \tag{D.4}$$

This transformation is equivalent to the following affine transformation of observation vector:

$$b_i(\boldsymbol{o}) = \mathcal{N}(\boldsymbol{o}; \boldsymbol{\zeta}'\boldsymbol{\mu}_i - \boldsymbol{\epsilon}', \boldsymbol{\zeta}'\boldsymbol{\Sigma}_i{\boldsymbol{\zeta}'}^\top) \tag{D.5}$$

$$= |\boldsymbol{\zeta}|\,\mathcal{N}(\boldsymbol{\zeta}\boldsymbol{o} + \boldsymbol{\epsilon}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{D.6}$$

$$= |\boldsymbol{\zeta}|\,\mathcal{N}(\boldsymbol{W}\boldsymbol{\xi}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{D.7}$$

$$p_i(d) = \mathcal{N}(d; \chi' m_i - \nu', \chi'\sigma_i^2\chi') \tag{D.8}$$

$$= |\chi|\,\mathcal{N}(\chi d + \nu; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{D.9}$$

$$= |\chi|\,\mathcal{N}(\boldsymbol{X}\boldsymbol{\phi}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{D.10}$$

where $\boldsymbol{\zeta} = {\boldsymbol{\zeta}'}^{-1}$ and $\boldsymbol{\epsilon} = {\boldsymbol{\zeta}'}^{-1}\boldsymbol{\epsilon}'$ are $L \times L$ matrix and L-dimensional vector, respectively, and $\chi = {\chi'}^{-1}$ and $\nu = {\chi'}^{-1}\nu'$ are scalar variables. $\boldsymbol{W} = [\boldsymbol{\zeta}, \boldsymbol{\epsilon}]$

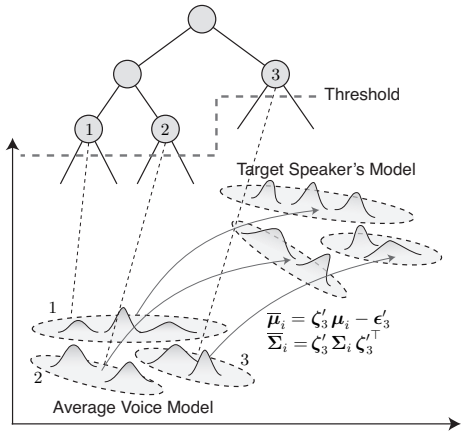


Figure D.1: Constrained Maximum Likelihood Linear Regression

and $\boldsymbol{X} = [\chi, \nu]$ are $L \times (L+1)$ and 1×2 transformation matrices, and $\boldsymbol{\xi} = [\boldsymbol{o}^\top, 1]^\top$ and $\boldsymbol{\phi} = [d, 1]^\top$ are $(L+1)$-dimensional and 2-dimensional vectors, respectively. Re-estimation formulas based on Baum-Welch algorithm of the transformation matrices can be derived as follows:

$$\overline{\boldsymbol{w}}_l = (\alpha \boldsymbol{p}_l + \boldsymbol{y}_l) \boldsymbol{G}_l^{-1} \tag{D.11}$$

$$\overline{\boldsymbol{X}} = (\beta \boldsymbol{q} + \boldsymbol{z}) \boldsymbol{K}^{-1} \tag{D.12}$$

where $\boldsymbol{w}_l$ is the l-th row vector of $\boldsymbol{W}$, and $\boldsymbol{q}_l = [0\, \boldsymbol{c}_l^\top]^\top$ and $\boldsymbol{q} = [0\, 1]^\top$ are $(L+1)$-dimensional and 2-dimensional vectors, respectively. It is note that $\boldsymbol{c}_l$ is l-th cofactor row vector of $\boldsymbol{W}$. Then $(L+1)$-dimensional vector $\boldsymbol{y}_l$, $(L+1) \times (L+1)$ matrix $\boldsymbol{G}_l$, 2-dimensional vector $\boldsymbol{z}$, and 2×2 matrix $\boldsymbol{K}$ are given by

$$\boldsymbol{y}_l = \sum_{r=1}^{R_b} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \, \frac{1}{\Sigma_r(l)} \, \mu_r(l) \sum_{s=t-d+1}^{t} \boldsymbol{\xi}_s^\top \tag{D.13}$$

$$\boldsymbol{G}_l = \sum_{r=1}^{R_b} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \, \frac{1}{\Sigma_r(l)} \sum_{s=t-d+1}^{t} \boldsymbol{\xi}_s \, \boldsymbol{\xi}_s^\top \tag{D.14}$$

$$\boldsymbol{z} = \sum_{r=1}^{R_p} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \, \frac{1}{\sigma_r^2} \, m_r \, \boldsymbol{\phi}_s^\top \tag{D.15}$$

$$\boldsymbol{K} = \sum_{r=1}^{R_p} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) \, \frac{1}{\sigma_r^2} \, \boldsymbol{\phi}_s \, \boldsymbol{\phi}_s^\top, \tag{D.16}$$

where $\Sigma_r(l)$ is the l-th diagonal element of diagonal covariance matrix $\boldsymbol{\Sigma}_r$, and $\mu_r(l)$ is the l-th element of the mean vector $\boldsymbol{\mu}_r$. Note that α and β are scalar values which satisfy the following quadratic equations:

$$\alpha^2 \boldsymbol{p}_l \boldsymbol{G}_l^{-1} \boldsymbol{p}_l^\top + \alpha \boldsymbol{p}_l \boldsymbol{G}_l^{-1} \boldsymbol{y}_l^\top - \sum_{r=1}^{R_b} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r)\, d = 0 \tag{D.17}$$

$$\beta^2 \boldsymbol{q} \boldsymbol{K}^{-1} \boldsymbol{q}^\top + \beta \boldsymbol{q} \boldsymbol{K}^{-1} \boldsymbol{z}^\top - \sum_{r=1}^{R_p} \sum_{t=1}^{T} \sum_{d=1}^{t} \gamma_t^d(r) = 0. \tag{D.18}$$

The CMLLR adaptation algorithm tunes not only mean values but also the range of the variation to a new speaker. Because the range of the variation is one of the important factors for F0 and duration, this algorithm would conduct more appropriate adaptation of prosodic information.

D.3 Experiments

D.3.1 Experimental Conditions

To compare and verify the effectiveness of each speaker adaptation algorithm, we conducted several objective and subjective evaluation tests for the synthetic speech using each speaker adaptation algorithm. Speech database for the following experiments contains 7 male and 5 female speakers' speech samples. Each speaker uttered a set of 503 phonetically balanced sentences taken from the ATR Japanese speech database. We chose 4 males and 4 females as training speakers for the average voice model, and used the rest of 3 males and 1 female as target speakers of the speaker adaptation. In the modeling of synthesis units, we used 42 phonemes, including silence and pause and took the phonetic and linguistic contexts [49] into account.

Speech signals were sampled at a rate of 16kHz and windowed by a 25ms Blackman window with a 5ms shift. The feature vectors consisted of 25 mel-cepstral coefficients including the zeroth coefficient, logarithm of F0, and their delta and delta-delta coefficients. We used 5-state left-to-right HSMMs without skip path. The gender-dependent and independent average voice models were separately trained using 1800 and 3600 sentences, respectively,

450 sentences for each training speaker. In the training stage of the average voice models, shared-decision-tree-based context clustering algorithm and speaker adaptive training [49] [47] were applied to normalize influence of speaker differences among the training speakers and train appropriate average voice models. Note that all the average voice models have the same topology and the number of distributions based on the shared-decision-trees.

We then adapted the average voice model to the target speaker using adaptation data whose sentences were included in the training sentences. In the all adaptation algorithms except SBR, multiple transformation parameters were estimated based on the shared-decision-trees constructed in the training stage of the average voice models. The tuning parameters for each adaptation algorithm, the thresholds to control the number of transformation parameters and hyper-parameters of the MAP estimation, were determined based on preliminary objective experimental results. The average voice model used as an initial model were also determined based on the preliminary objective experimental results. The gender-dependent average voice models were used for 2 male and 1 female speakers and gender-independent average voice model was used for the rest of a male speaker. ESAT adaptation used both the gender-dependent and gender-independent average voice models as the initial models.

D.3.2 Objective Evaluations

Firstly, we calculated the target speakers' average mel-cepstral distance and root-mean-square (RMS) error of logarithmic F0 as the objective evaluations for each speaker adaptation algorithm. The number of the adaptation sentences ranged from three to a hundred. Fifty test sentences were used for evaluation, which were included in neither training nor adaptation data. For the distance calculation, state duration of each model was adjusted after Viterbi alignment with the target speaker's real utterance.

Figure D.2 shows the target speakers' average mel-cepstral distance between spectra generated from each model and obtained by means of analyzing target speaker's real utterance, and the RMS logarithmic F0 error between generated logarithmic F0 and that extracted from target speaker's real utter-

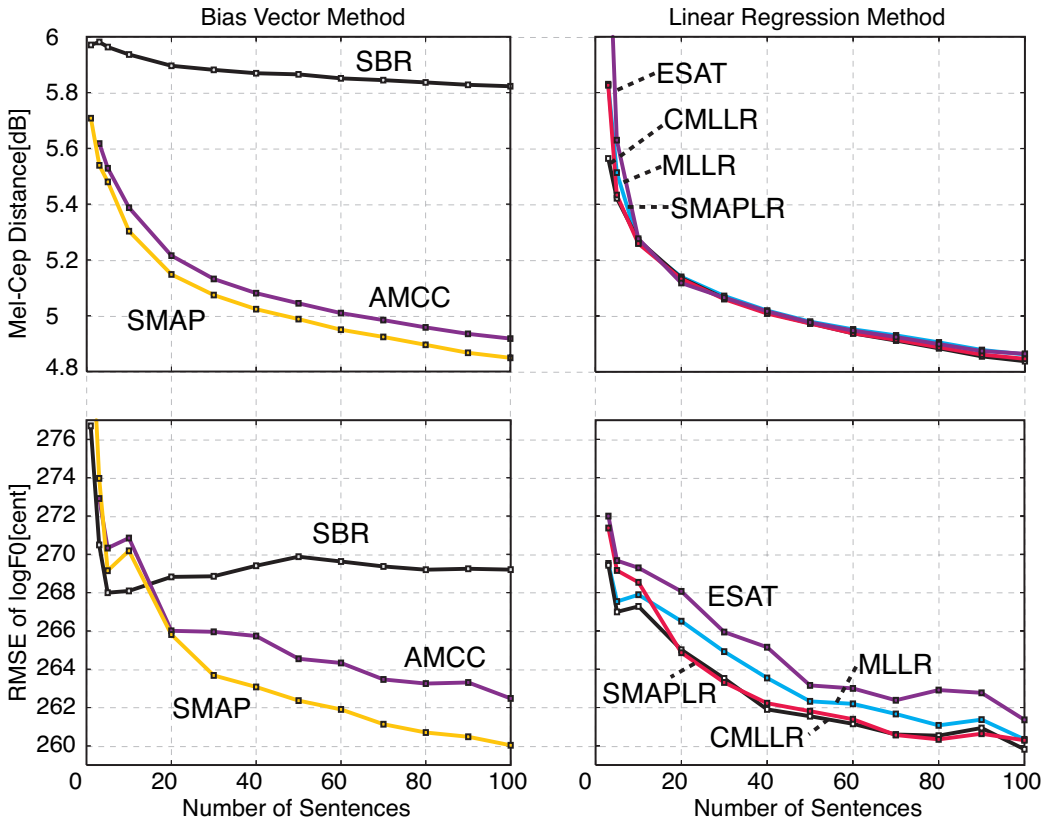


Figure D.2: Objective evaluation of speaker adaptation algorithms.

ance. In the distance calculation, silence and pause regions were eliminated. And since F0 value is not observed in the unvoiced region, the RMS logarithmic F0 error was calculated in the region where both generated F0 and real F0 were voiced. From this figure, it can be seen that CMLLR adaptation to tune both mean and variance also have a beneficial effect on the improvements of F0 part. On the other hand, ESAT adaptation does not provide the improvements in both spectrum and F0 parts.

D.4 Conclusions

This chapter has described comparative study of several speaker adaptation algorithms including multiple linear regression and constrained MLLR From

the results of the objective evaluation test, we have evaluated the advantages and effectiveness of the CMLLR adaptation algorithms.

Acknowledgments

First, I would like to express my thanks to Professor Takao Kobayashi, Tokyo Institute of Technology, for all of his support, encouragement, and guidance. Also, I would like to thank to Professor Yoshinori Hatori, Professor Hideo Maejima, Professor Keiji Uchikawa, Professor Makoto Sato, and Professor Hiroshi Nagahashi of Tokyo Institute of Technology, and Professor Keikichi Hirose of The University of Tokyo for their kind suggestions.

I would like to also thank Professor Keiichi Tokuda of Nagoya Institute of Technology. His substantial help in my work are deeply appreciated. In addition to this, I have benefited greatly from interaction with members of the Kobayashi Laboratory at Tokyo Institute of Technology, and members of the Tokuda Laboratory at Nagoya Institute of Technology over the years. There are too many people to mention individually, but I must thank Takashi Masuko, Masatsune Tamura (currently with Toshiba Corporation), Heiga Zen, Yoshihiko Nankaku of Nagoya Institute of Technology, Tomoki Toda of Nara Institute of Science and Technology, Hisashi Kawai of KDDI R & D Laboratories, Minoru Tsuzaki of Kyoto City University of Arts, Toshio Hirai, Nobuyuki Nishizawa, Jinfu NI of ATR Spoken Language Translation Research Laboratories, Sadao Hiroya of NTT Communication Science Laboratories, Koji Onishi of NEC Corporation, Naotake Niwase of NTT Data Corporation and Makoto Tachibana of the Kobayashi Laboratory. Without their help, I could not possibly have completed this work.

Finally, I would like to give my special thanks to my family for all their support over the years.

Bibliography

[1] E. Moulines and F. Charpentier, "Pitch synchronous waveform processing techniques for text-to-speech synthesis using diphones," Speech Communication, vol.9, no.5–6, pp.453–467, 1990.

[2] A.W. Black and N. Cambpbell. Optimising selection of units from speech database for concatenative synthesis. In *Proc. EUROSPEECH-95*, pages 581–584, September 1995.

[3] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. *J. Acoust. Soc. Jpn. (E)*, 11(2):71–76, 1990.

[4] K. Ito and S. Saito, "Effects of acoustical feature parameters of speech on perceptual identification of speaker," IECE Trans. A, vol.J65-A, no.1, pp.101–108, Jan. 1982 (in Japanese).

[5] N. Higuchi and M. Hashimoto, "Analysis of acoustic features affecting speaker identification," Proc. EUROSPEECH-95, pp.435–438, Sep. 1995.

[6] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Speaker adaptation for HMM-based speech synthesis system using MLLR. In *The Third ESCA/COCOSDA Workshop on Speech Synthesis*, pages 273–276, November 1998.

[7] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Adaptation of pitch and spectrum for HMM-based speech synthesis using MLLR. In *Proc. ICASSP 2001*, pages 805–808, May 2001.

[8] M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. Text-to-speech synthesis with arbitrary speaker's voice from average voice. In *Proc. EUROSPEECH 2001*, pages 345–348, September 2001.

[9] T. Masuko, K. Tokuda, T. Kobayashi, and S. Imai. Speech synthesis using HMMs with dynamic features. In *Proc. ICASSP-96*, pages 389–392, May 1996.

[10] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis. In *Proc. EUROSPEECH-99*, pages 2374–2350, September 1999.

[11] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden Markov models for speech recognition*, Edinburgh University Press, 1990.

[12] L. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*, Prentice-Hall, Englewood Cliffs, N. J., 1993.

[13] S. Young, G. Everman, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book Version 3.2.1*, December 2002.

[14] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis," IEICE Trans. D-II, vol.J83-D-II, no.11, pp.2099–2107, Nov. 2000 (in Japanese).

[15] K. Tokuda, T. Kobayashi, and S. Imai. Speech parameter generation from HMM using dynamic features. In *Proc. ICASSP-95*, pages 660–663, May 1995.

[16] K. Tokuda, T. Masuko, T. Kobayashi, and S. Imai. An algorithm for speech parameter generation from hmm using dynamic features. *J. Acoust. Soc. Japan (J)*, 53(3):192–200, March 1997. (in Japanese).

[17] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorigthms for HMM-based speech synthesis. In *Proc. ICASSP 2000*, pages 1315–1318, June 2000.

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

[19] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Hidden Markov models based on multi-space probability distribution for pitch pattern modeling. In *Proc. ICASSP-99*, pages 229–232, March 1999.

[20] K. Tokuda, T. Masuko, N. Miyazaki, and T. Kobayashi. Multi-space probability distribution hmm. *IEICE Trans. Inf. & Syst.*, J83-D-II(7):1579–1589, July 2000. (in Japanese).

[21] T. Masuko, K. Tokuda, N. Miyazaki, and T. Kobayashi. Pitch pattern generation using multi-space probability distribution HMM. *IEICE Trans. Inf. & Syst.*, J83-D-II(7):1600–1609, July 2000. (in Japanese).

[22] S. J. Young, J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," Proc. ARPA Human Language Technology Workshop, pp.307–312, Mar. 1994.

[23] K. Shinoda and T. Watanabe. MDL-based context-dependent subword modeling for speech recognition. *J. Acoust. Soc. Japan (E)*, 21:79–86, March 2000.

[24] S.J. Young, J.J. Odell, and P.C. Woodland. Tree-based state tying for high accuracy acoustic modeling. In *Proc. ARPA Human Language Technology Workshop*, pages 307–312, March 1994.

[25] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Duration modeling for HMM-based speech synthesis. In *Proc. ICSLP-98*, pages 29–32, December 1998.

[26] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (MLSA) filter for speech synthesis," IECE Trans. A, vol.J66-A, no.2, pp.122–129, Feb. 1983 (in Japanese).

[27] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai. An adaptive algorithm for mel-cepstral analysis of speech. In *Proc. ICASSP-92*, pages 137–140, March 1992.

[28] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," Proc. ICASSP-88, pp.655–658, Apr. 1988.

[29] M. Hashimoto and N. Higuchi, "Spectral mapping method for voice conversion using speaker selection and vector field smoothing techniques," IEICE Trans. D-II, vol.J80-D-II, no.1, pp.1–9, Jan. 1997 (in Japanese).

[30] Y. Stylianou and O. Cappé, "A system for voice conversion based on probabilistic classification and a harmonic plus noise model," Proc. ICASSP-98, pp.281–284, May 1998.

[31] M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda, "Speaker adaptation of pitch and spectrum for HMM-based speech synthesis," IEICE Trans. D-II, vol.J85-D-II, no.4, pp.545–553, Apr. 2002 (in Japanese).

[32] C.J. Leggetter and P.C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9(2):171–185, 1995.

[33] K. Tokuda, T. Kobayashi, T. Fukada, H. Saito, and S. Imai. Spectral estimation of speech based on mel-cepstral representation. In *IEICE Trans. Fundamentals (Japanese Edition), vol.J74-A, no.8*, pages 1240–1248, August 1991.

[34] Entropic Research Laboratory Inc. *ESPS Programs Version 5.0*, 1993.

[35] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul. A compact model for speaker-adaptive training. In *Proc. ICSLP-96*, pages 1137–1140, October 1996.

[36] J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. A context clustering technique for average voice models. *IEICE Trans. Inf. & Syst.*, E86-D(3):534–542, March 2003.

[37] E. Eide and H. Gish. A parametric approach to vocal tract length normalization. In *Proc. ICASSP-96*, pages 346–349, May 1996.

[38] L. Lee and R.C. Rose. Speaker normalization using efficient frequency warping procedures. In *Proc. ICASSP-96*, pages 353–356, May 1996.

[39] P. Zhan and M. Westohal. Speaker normalization based on frequency warping. In *Proc. ICASSP-97*, pages 1039–1043, April 1996.

[40] T. Tanaka, T. Kobayashi, D. Arifianto, and T. Masuko. Fundamental frequency estimation based on instantaneous frequency amplitude spectrum. In *Proc. ICASSP 2002*, pages 329–332, May 2002.

[41] J.D. Ferguson. Variable duration models for speech. In *Symp. on the Application of Hidden Markov Models to Text and Speech*, pages 143–179, 1980.

[42] S.E. Levinson. Continuously variable duration hidden Markov models for automatic speech recognition. *Computer Speech and Language*, 1(1):29–45, 1986.

[43] M.J. Russell and R.K. Moore. Explicit modelling of state occupancy in hidden Markov models for automatic speech recognition. In *Proc. ICASSP-85*, pages 5–8, March 1985.

[44] M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi. Performance evaluation of style adaptation for hidden semi-markov model. In *Proc. EUROSPEECH 2005*, pages 2805–2808, September 2005.

[45] M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi. A style adaptation technique for speech synthesis using HSMM and suprasegmental features. *IEICE Trans. Information and Systems*, March 2006. (to appear).

[46] J. Yamagishi, T. Masuko, and T. Kobayashi. MLLR adaptation for hidden semi-Markov model based speech synthesis. In *Proc. ICSLP 2004*, pages 1213–1216, October 2004.

[47] J. Yamagishi and T. Kobayashi. Adaptive training for hidden semi-Markov model. In *Proc. ICASSP 2005*, pages 365–368, March 2005.

[48] J. Yamagishi, M. Tachibana, T. Masuko, and T. Kobayashi. Speaking style adaptation using context clustering decision tree for HMM-based speech synthesis. In *Proc. ICASSP 2004*, pages 5–8, May 2004.

[49] J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi. A training method of average voice model for HMM-based speech synthesis. *IEICE Trans. Fundamentals*, E86-A(8):1956–1963, August 2003.

[50] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura. Hidden semi-Markov model based speech synthesis. In *Proc. ICSLP 2004*, pages 1393–1396, October 2004.

[51] P. McCullagh and J.A. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.

[52] M. Rahim and B.H. Juang. Signal bias removal by maximum likelihood estimation for robust telephone speech recognition. *IEEE Trans. Speech Audio Processing*, 4:19–30, January 1996.

[53] L. Neumeyer, V. Digalakis, and M. Weintraub. Training issues and channel equalization techniques for the construction of telephone acoustic models using a high-quality speech corpus. *IEEE Trans. Speech Audio Processing*, 2:590–597, October 1994.

[54] K. Shinoda and T. Watanabe. Speaker adaptation with autonomous control using tree structure. In *Proc. EUROSPEECH-95*, pages 1143–1146, September 1995.

[55] K. Shinoda and T. Watanabe. Speaker adaptation with autonomous model complexity control by MDL principle. In *Proc. ICASSP-96*, pages 717–720, May 1996.

[56] K. Shinoda and C.H. Lee. A structural Bayes approach to speaker adaptation. *IEEE Trans. Speech Audio Process.*, 9:276–287, March 2001.

[57] O. Shiohan, T.A. Myrvoll, and C.H. Lee. Structural maximum a posteriori linear regression for fast HMM adaptation. *Computer Speech and Language*, 16(3):5–24, 2002.

[58] V. Digalakis and L. Neumeyer. Speaker adaptation using combined transformation and Bayesian methods. *IEEE Trans. Speech Audio Processing*, 4:294–300, July 1996.

[59] J.T. Chien, H.C. Wang, and C.H. Lee. Improved Bayesian learning of hidden Markov models for speaker adaptation. In *Proc. ICASSP-97*, pages 1027–1030, April 1997.

[60] M.J.F. Gales. Multiple-cluster adaptive training schemes. In *Proc. ICASSP 2001*, pages 361–364, May 2001.

[61] V. Digalakis, D. Rtischev, and L. Neumeyer. Speaker adaptation using constrained reestimation of Gaussian mixtures. *IEEE Trans. Speech Audio Processing*, 3:357–366, September 1995.

[62] M.J.F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2):75–98, 1998.

[63] *Emotional speech synthesis: a review*, September 2001.

[64] M. Abe. Speaking styles : Statistical analysis and synthesis by a text-to-speech system. In J.P.H. van Santen, R.W. Sproat, J.P. Olive, and J. Hirschberg, editors, *Progress in Speech Synthesis*, pages 495–510. Springer, 1997.

[65] J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi. Modeling of various speaking styles and emotions for HMM-based speech synthesis. In *Proc. EUROSPEECH 2003*, pages 2461–2464, September 2003.

[66] Speaking Style & Emotional Speech Database (SS2003) in Prosodic Corpus, Scientific Research of Priority Areas "Prosody and Speech Processing," 2003.

[67] J. Isogai, J. Yamagishi, and T. Kobayashi. Model adaptation and adaptive training using ESAT algorithm for HMM-based speech synthesis. In *Proc. EUROSPEECH 2005*, pages 2597–2600, September 2005.

List of Publications

Publications Related to This Thesis

Journal

1. J. Yamagishi and T. Kobayashi,
   "Simultaneous Speaker Adaptation Algorithm of Spectrum, Fundamental Frequency and Duration for HMM-based Speech Synthesis,"
   IEICE Trans. Information and Systems. (in preparation)

2. J. Yamagishi, Y. Nakano, K. Ogata, J. Isogai, and T. Kobayashi,
   "A Unified Speech Synthesis Method Using
   HSMM-Based Speaker Adaptation and MAP Modification",
   IEICE Trans. Information and Systems. (in preparation)

3. J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi,
   "Acoustic Modeling of Speaking Styles and Emotional Expressions in HMM-based Speech Synthesis,"
   IEICE Trans. Information and Systems,
   E88-D, vol.3, pp.503–509, March 2005.

4. J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi,
   "A Training Method of Average Voice Model for HMM-based Speech Synthesis",
   IEICE Trans. Fundamentals,
   E86-A, no.8, pp.1956–1963, August 2003.

5. J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi,
   "A Context Clustering Technique for Average Voice Models",
   IEICE Trans. Information and Systems,

E86-D, no.3, pp.534–542, March 2003

International Conference

1. J. Yamagishi, K. Ogata, Y. Nakano, J. Isogai, and T. Kobayashi,
"HSMM-based Model Adaptation Algorithms
for Average-Voice-based Speech Synthesis",
Proc. ICASSP 2006, May 2006 (submit).

2. J. Yamagishi, and T. Kobayashi,
"Adaptive Training for Hidden Semi-Markov Model",
Proc. ICASSP 2005, vol.I, pp.365–368, March 2005.

3. J. Yamagishi, T. Masuko, and T. Kobayashi,
"MLLR Adaptation for Hidden Semi-Markov Model Based Speech Synthesis",
Proc. ICSLP 2004, vo.II, pp.1213–1216, October 2004.

4. J. Yamagishi, M. Tachibana, T. Masuko, and T. Kobayashi,
"Speaking Style Adaptation Using Context Clustering Decision Tree for HMM-based Speech Synthesis",
Proc. ICASSP 2004 , vol.I, pp.5–8, May 2004.

5. J. Yamagishi, T. Masuko, and T. Kobayashi,
"HMM-based Expressive Speech Synthesis – Towards TTS with Arbitrary Speaking Styles and Emotions,"
Special Workshop in Maui (SWIM) , January 2004.

6. J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi,
"Modeling of Various Speaking Styles and Emotions for HMM-based Speech Synthesis",
Proc. EUROSPEECH 2003, vol.III, pp.2461–2464, September 2003.

7. J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi,
"A Training Method for Average Voice Model Based on Shared Decision Tree Context Clustering and Speaker Adaptive Training",
Proc. ICASSP 2003, vol.I, pp.716–719, April 2003.

8. J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda, and T. Kobayashi,
"A Context Clustering Technique for Average Voice Model in HMM-based

Speech Synthesis",
Proc. ICSLP 2002, vol.1, pp.133–136, September 2002.

IEICE Technical Report

1. J. Yamagishi, H. Kawai, T. Hirai, T. Kobayashi,
"Phone Duration Modeling Based on Ensemble Learning,"
IEICE Technical Report,
vol.105, no.253, SP2005-53, pp.7–12, August 2005. (in Japanese).

2. J. Yamagishi, T. Masuko, and T. Kobayashi,
"A Study on MLLR-based Style Adaptation in HSMM-based Speech Synthesis,"
IEICE Technical Report,
vol.104, no.252, SP2004-49, pp.13–18, August 2004. (in Japanese).

3. J. Yamagishi, T. Masuko, K. Tokuda and T. Kobayashi,
"Speaker Adaptation Using Context Clustering Decision Tree for HMM-based Speech Synthesis,"
IEICE Technical Report,
vol.103, no.264, SP2003-79, pp.31–36, August 2003. (in Japanese).

4. J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda and T. Kobayashi,
"A Study on Context Clustering Techniques and Speaker Adaptive Training for Average Voice Model,"
IEICE Technical Report,
vol.102, no.292, SP2002-72, pp.5–10, August 2002. (in Japanese).

5. J. Yamagishi, M. Tamura, T. Masuko, K. Tokuda and T. Kobayashi,
"A Study on A Context Clustering Technique for Average Voice Models,"
IEICE Technical Report,
vol.102, no.108, SP2002-28, pp.25–30, May 2002. (in Japanese).

ASJ Meeting

1. J. Yamagishi and T. Kobayashi,
"Hidden Semi-Markov Model based Adaptive Training,"
ASJ Spring meeting, 1-1-17, pp.187–188, March 2005 (in Japanese).

2. J. Yamagishi, T. Masuko, and T. Kobayashi,
"A Study on HSMM-based MLLR Adaptation",
ASJ Autumn meeting, 3-2-8, pp.331–332, September 2004 (in Japanese).

3. J. Yamagishi, T. Masuko, and T. Kobayashi,
"A Study on State Duration Modeling Using Lognormal Distribution for HMM-based Speech Synthesis,"
ASJ Spring meeting, 1-7-7, pp.225–226, March 2004 (in Japanese).

4. J. Yamagishi, T. Masuko, T. Kobayashi, and K. Tokuda,
"A Study on Speaker Adaptation Technique Using Context Clustering Decision Tree,"
ASJ Autumn meeting, 1-8-16, pp.213–214, September 2003 (in Japanese).

5. J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi,
"Speaking Styles Control in HMM-based Speech Synthesis,"
IPSJ meeting, 5T7B-5, pp.511–514, March 2003 (in Japanese).

6. J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi,
"A Study on Adaptation Technique for Speech Synthesis with Arbitarary Speaker's Voice and Various Speaking Styles",
ASJ Spring meeting, 1-6-25, pp.271–272, March 2003 (in Japanese).

7. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda,
"A Study on Training Methods of Average Voice Models for Speaker Adaptation,"
ASJ Autumn meeting, 3-10-12, pp.351–352, September 2002 (in Japanese).

8. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda,
"Evaluation of Training Methods of Average Voice Models for Speaker Adaptation,"
ASJ Autumn meeting, 3-10-13, pp.353–354, September 2002 (in Japanese).

9. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda,
"A Study on Construction Techniques of Decision Tree for HMM-based Speech Synthesis,"
ASJ Spring meeting, 1-10-1, pp.231–232, March 2002 (in Japanese).

10. J. Yamagishi, M. Tamura, T. Masuko, T. Kobayashi, and K. Tokuda,
"A Study on Trainig Data of Average Voice Models for HMM-based Speech

Synthesis,"
ASJ Autumn meeting, 3-2-10, pp.323–324, Octorber 2001 (in Japanese).

Other Publications

Journal

1. M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi,
"A Style Adaptation technique for HMM-based Speech Synthesis using HSMM and Suprasegmental Features",
IEICE Trans. Information and Systems,
March 2006. (to appear)

2. N. Niwase, J. Yamagishi, and T. Kobayashi,
"Human Walking Motion Synthesis Having Desired Pace and Stride Length Based on HSMM",
IEICE Trans. Information and Systems,
E88-D, no.11, pp.2492–2499, November 2005.

3. M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi,
"Speech Synthesis with Various Emotional Expressions and Speaking Styles by Style Interpolation and Morphing",
IEICE Trans. Information and Systems,
E88-D, no.11, pp.2484–2491, November 2005.

International Conference

1. T. Nose, J. Yamagishi, and T. Kobayashi,
"Modeling and Classification of Emotional Speech Based on Multiple Regression Hidden Semi-Markov Model",
Proc. ICASSP 2006, May 2006 (submit).

2. T. Yamazaki, N. Niwase, J. Yamagishi, and T. Kobayashi,
"Human Walking Motion Synthesis Based on Multiple Regression Hidden Markov Model",
Proc. CW2005-LUAR2005, Nov. 2005.

3. M. Tachibana, J. Yamagishi, T. Masuko, and T. Kobayashi,

"Performance Evaluation of Style Adaptation
for Hidden Semi-Markov Model",
Proc. EUROSPEECH 2005, pp.2597–2600, Sept. 2005.

4. J. Isogai, J. Yamagishi, and T. Kobayashi,
"Model Adaptation and Adaptive Training
using ESAT Algorithm for HMM-based Speech Synthesis",
Proc. EUROSPEECH 2005, pp.2805–2808, Sept. 2005.

5. M. Tachibana, J. Yamagishi, K. Onishi, T. Masuko, and T. Kobayashi,
"HMM-based Speech Synthesis with Various Speaking Styles Using Model Interpolation",
Proc. Speech Prosody 2004 , pp.413–416, March 2004.

IEICE Technical Report

1. K. Kawashima, M. Tachibana, J. Yamagishi, and T. Kobayashi,
"Style Classification of Speech Based on MSD-HMM,"
IEICE Technical Report,
Dec. 2005 (in Japanese) (to appear).

2. M. Tachibana, J. Yamagishi, and T. Kobayashi,
"Performance Evaluation of Style Adaptation
for Hidden Semi-Markov Model Based Speech Synthesis,"
IEICE Technical Report,
vol.105, no.252, SP2005-51, pp.29–34, August 2005. (in Japanese).

3. J. Isogai, K. Ogata, Y. Nakano, J. Yamagishi, and T. Kobayashi,
"Model Adaptation and Adaptive Training Algorithms for Speech Synthesis with Diverse Speaker Characteristics,"
IEICE Technical Report,
vol.105, no.252, SP2005-50, pp.23–28, August 2005. (in Japanese).

4. T. Hirai, H. Kawai, T. Toda, J. Yamagishi, J. Ni, N. Nishizawa, M. Tsuzaki, K. Tokuda,
"XIMERA: A New Text-to-Speech from ATR based on Corpus-based Technologies"
IEICE Technical Report,
vol.105, no.98, SP2005-18, pp.37–42, May 2005. (in Japanese).

5. M. Tachibana, J. Yamagishi, K. Onishi, and T. Kobayashi,
   "HMM-based Speech Synthesis with Various Speaking Styles Using Model Interpolation and Adaptation,"
   IEICE Technical Report,
   vol.103, no.264, SP2003-80, pp.37–42, August 2003. (in Japanese).

ASJ Meeting

1. J. Yamagishi, H. Kawai, and T. Kobayashi,
   "Pause Prediction Algorithm Based on Naïve Markov Model,"
   ASJ Autumn meeting, 3-6-11, pp.349–350, September 2005 (in Japanese).

2. J. Yamagishi, H. Kawai, and T. Kobayashi,
   "A Study on Segmental Duration Modeling Using Ensemble Learning,"
   ASJ Autumn meeting, 3-2-1, pp.317–318, September 2004 (in Japanese).

3. K. Kawashima, J. Yamagishi, and T. Kobayashi,
   "A Study on Style Classification of Read Speech Based on MSD-HMM,"
   ASJ Autumn meeting, 1-P-24, pp.199–200, September 2005 (in Japanese).

4. Y. Yuji, K. Ogata, J. Isogai, J. Yamagishi, and T. Kobayashi,
   "A Comparative Study of Speaker Adaptation Algorithms
   for Average-Voice-Based Speech Synthesis,"
   ASJ Autumn meeting, 1-Q-11, pp.395–396, September 2005 (in Japanese).

5. T. Hirai, J. Yamagishi, H. Kawai, J. Ni, N. Nishizawa, K. Tokuda, M. Tsuzaki, and T. Toda,
   "Conversational Speech Synthesis by XIMERA"
   ASJ Autumn meeting, 2-6-6, pp.269–270, September 2005 (in Japanese).

6. T. Nose, J. Yamagishi, and T. Kobayashi,
   "A Study on Style Control Technique for Speech Synthesis Using Multiple Regression HSMM,"
   ASJ Autumn meeting, 2-6-13, pp.287–288, September 2005 (in Japanese).

7. M. Tachibana, J. Yamagishi, and T. Kobayashi,
   "A Comparative Study of Style Adaptation Algorithms
   for Expressive Speech Synthesis,"
   ASJ Autumn meeting, 2-6-14, pp.289-290, September 2005 (in Japanese).

8. J. Isogai, J. Yamagishi, and T. Kobayashi,
"Performance Evaluation of ESAT Algorithm for HMM-based Speech Synthesis,"
ASJ Autumn meeting, 3-6-23, pp.373–374, September 2005 (in Japanese).

9. M. Tachibana, J. Yamagishi, T. Masuko and T. Kobayashi,
"A Technique for Controling Speaking Style and Emotional Expression Using Model Interpolation,"
ASJ Spring meeting, 1-1-19, pp.191–192, March 2005 (in Japanese).

10. D. Nomura, J. Yamagishi, and T. Kobayashi,
"A Study on Stopping Criteria for Decision-Trees in HMM-based Speech Synthesis,"
ASJ Spring meeting, 3-P-26, pp.291-292, March 2005 (in Japanese).

11. J. Isogai, J. Yamagishi, and T. Kobayashi,
"A Study on Model Adaptation and Adaptive Training Using ESAT Algorithm in HMM-Based Speech Synthesis",
ASJ Spring meeting, 1-1-18, pp.189-190, March 2005 (in Japanese).

12. N. Niwase, J. Yamagishi, and T. Kobayashi,
"Generation of Walking Movements with an Arbitrarily Prescribed Pace,"
Proceedings of the 2005 IEICE General Conference, D-12-118, pp.268, March 2005 (in Japanese).

13. M. Tachibana, J. Yamagishi, T. Masuko and T. Kobayashi,
"Performance Evaluation of Style Adaptation in HSMM-based Speech Synthesis",
ASJ Autumn meeting, 3-2-9, pp.333–334, September 2004 (in Japanese).

14. M. Tachibana, J. Yamagishi, T. Masuko and T. Kobayashi,
"Performance Evaluation of Style Adaptation Using Context Clustering Decision Tree,"
ASJ Spring meeting, 1-7-22, pp.255–256, March 2004 (in Japanese).

15. D. Sanno, J. Yamagishi, T. Masuko and T. Kobayashi,
"A Study on Style Adaptation Using Structural MAPLR in HMM-based Speech Synthesis",
ASJ Spring meeting, 1-7-23, pp.257–258, March 2004 (in Japanese).

16. M. Tachibana, J. Yamagishi, T. Masuko and T. Kobayashi,
"A Study on Speaking Style Adaptation in HMM-based Speech Synthesis,"
ASJ Autumn meeting, 1-8-29, pp.239–240, September 2003 (in Japanese).

Other

1. J. Yamagishi and H. Kawai,
"Segmental Duration Modeling Using Ensemble Learning,"
ATR Technical Report, TR-SLT-0079, August 2004.