Statistical Parametric Speech Synthesis Based on Gaussian Process Regression

Tokmoki Koriyama, *Student Member, IEEE,* Takashi Nose, *Member, IEEE,*
and Takao Kobayashi, *Senior Member, IEEE*

***Abstract*—This paper proposes a statistical parametric speech synthesis technique based on Gaussian process regression (GPR). The GPR model is designed for directly predicting frame-level acoustic features from corresponding information on frame context that is obtained from linguistic information. The frame context includes the relative position of the current frame within the phone and articulatory information and is used as the explanatory variable in GPR. Here, we introduce cluster-based sparse Gaussian processes (GPs), i.e., local GPs and partially independent conditional (PIC) approximation, to reduce the computational cost. The experimental results for both isolated phone synthesis and full-sentence continuous speech synthesis revealed that the proposed GPR-based technique without dynamic features slightly outperformed the conventional hidden Markov model (HMM)-based speech synthesis using minimum generation error training with dynamic features.**

***Index Terms*—statistical speech synthesis, Gaussian process regression, nonparametric Bayesian model, sparse Gaussian processes, partially independent conditional (PIC) approximation**

I. Introduction

IN corpus-based statistical speech synthesis, statistical parametric speech synthesis based on hidden Markov model (HMM) [1] has been widely studied [2]. In HMM-based speech synthesis, observation vector sequences consisting of acoustic features are modeled using HMMs with hidden state sequences, and speech parameters are directly generated from a set of trained HMMs for given context labels [3]. The acoustic characteristics of each speech synthesis unit are represented at the segmental and supra-segmental levels by using context-dependent models where phonetic and prosodic contextual factors are taken into account.

Although the HMM-based speech synthesis can create synthetic speech that well reflects most acoustic characteristics of training data, there are two major problems. First, HMM essentially assumes the stationarity of output features within a discrete state while the characteristics of actual acoustic features change even within a state. To model such characteristics, dynamic features are generally incorporated as the acoustic features. However, the dynamic parameters of each state only represent the average dynamic property of the segment associated with the state, which is not always appropriate to model the variation of the segment. The second problem is generalization in the model training using decision-tree-based context clustering. The states of context-dependent HMMs are clustered using decision trees, and states in each leaf node are tied to a single state [4]. As a result, the total number of states is reduced to the total number of leaf nodes. Although this improves the estimation accuracy of model parameters and enables model parameters to be predicted for unseen contexts, the resulting number of model parameters is limited and contextual diversity decreases.

Several techniques have been proposed to alleviate the quality degradation caused by the above two problems. For the first problem, a minimum generation error (MGE) training [5], trajectory HMM [6], and autoregressive HMM [7] were proposed. In the MGE training and trajectory HMM, the explicit relationship between static and dynamic features is incorporated in the model training. The autoregressive HMM introduced the dependency of the observed static feature on not only the state but also the past observations. Rich context modeling [8], [9] is a technique for alleviating the second problem of the over-smoothing effect with the parameter-tying process. In this approach, the optimum untied HMM sequence for input context labels are searched for by using conventional tied HMMs as guiding models. The subjective quality is expected to be improved when there is a sufficient amount of training data and the contexts of training data adequately cover those of the input texts.

In recent years, novel approaches using Gaussian processes (GPs) have been proposed for speech processing, such as speech enhancement [10], voice conversion [11], phoneme classification [12], and acoustic modeling [13]. Henter et al. [13] attempted to solve the problem of state discreteness by extending discrete states to continuous variables of a latent space where GP was used for a frame-level function that transformed the latent space variables into acoustic features. They used a Gaussian process dynamical model (GPDM) to express the latent space. However, it is not easy to apply GPDM to text-to-speech directly because of the difficulty in correlating latent space variables with the linguistic information of a given input sentence to be synthesized.

In this paper, we propose a speech synthesis technique based on the Gaussian process regression (GPR) [14] to overcome the limitations of HMM-based synthesis. GPs are known to be nonparametric Bayesian models where “nonparametric” means that model complexity expands as increase data size increases. That is, the variations within a segment and among

A part of this work was supported by JSPS Grant-in-Aid for Scientific Research 24300071 and 25540065.
T. Koriyama, T. Nose, and T. Kobayashi are with the Department of Information Processing, Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, 226-8502, Japan. (e-mail: koriyama.t.aa@m.titech.ac.jp; takashi.nose@ip.titech.ac.jp; takao.kobayashi@ip.titech.ac.jp).

segments are modeled more appropriately using a sufficient number of parameters than HMM-based synthesis. Another advantage of GPs is the robust parameter estimation due to Bayesian inference that reduces the problem of over-fitting. Since GPs involve a kernel method, various kinds of data can be used as input variables by defining the kernel function of respective samples [15]. Moreover, the GPR, as well as other types of regression models, can directly represent the relationship between linguistic and acoustic features of corresponding frames without using parameter tying by decision tree clustering, which is inevitable in the HMM-based speech synthesis.

Although the proposed technique assumes GP on a frame-level function in the same way as that by Henter et al. [13], it differs in that the function transforms frame-level information obtained from linguistic information instead of latent space variables. Here, we define a combined kernel including the kernels for position and phone contexts. In addition, we incorporate approximation techniques into GPR-based speech parameter modeling to reduce the computational cost in calculating the covariance matrices.

The rest of this paper is organized as follows: Section II introduces the proposed framework based on GPR for isolated phone segments. Section III describes the basic performance of the proposed framework using isolated phone segments. In Section IV, we extend the proposed framework to continuous speech synthesis of full sentences. Section V presents the objective and subjective evaluation of the extended framework, where the proposed technique is compared with the HMM-based technique using MGE training. Finally we discuss the remaining issues and plans for improvement in Section VI and conclude this study in Section VII.

II. Speech synthesis based on Gaussian process regression for isolated phones

This section briefly describes the basic theory of general GPR [14] and then presents the framework of GPR-based speech synthesis for a small amount of speech data, i.e., isolated phone segments. A frame context kernel is designed as an input variable of the GPR to represent frame-level acoustic features.

A. Gaussian process for regression

Suppose that we have a training data set, $\mathcal{D} = \{(\mathbf{x}_n, y_n) | n = 1, \ldots, N\}$, and a test data set, $\mathcal{D}_T = \{(\mathbf{x}_t, y_t) | t = 1, \ldots, T\}$, where $\mathbf{x}_n$ is a column vector consisting of explanatory (input) variables, and y_n is an output scalar variable. We assume that y_n is given by

$$y_n = f(\mathbf{x}_n) + \epsilon \tag{1}$$

where $f(\mathbf{x}_n)$ is a noise-free latent function value and ϵ represents the Gaussian noise of $\epsilon \sim \mathcal{N}(\epsilon; 0, \sigma^2)$. Let $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top$ and $\mathbf{y} = [y_1, \ldots, y_N]^\top$ be matrix forms of all input and output variables of training data and $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_N)]^\top$ be the latent function values of the training data. We define $\mathbf{X}_T$, $\mathbf{y}_T$, and $\mathbf{f}_T$ as matrix forms for test data in the same way as the training data.

If output variables are normalized to zero mean and $f(\mathbf{x}_i)$ is a Gaussian process, the GP prior is given by

$$p(\mathbf{y}|\mathbf{X}) = \mathcal{N}(\mathbf{y}; \mathbf{0}, \mathbf{K}_N + \sigma^2 \mathbf{I}) \tag{2}$$

$\mathbf{K}_N$ is a covariance matrix (Gram matrix) of the training data whose element is given by

$$K_{mn} = k(\mathbf{x}_m, \mathbf{x}_n) \qquad m = 1 \ldots N, \; n = 1 \ldots N \tag{3}$$

and $k(\mathbf{x}_m, \mathbf{x}_n)$ is a kernel (or covariance) function.

The main goal of GPR is to infer the continuous distributions of output variables of test data, $\mathbf{y}_T$, given new input vectors $\mathbf{X}_T$. The joint distribution on the function values, $\mathbf{f}$ and $\mathbf{f}_T$, of the training and test data is given by

$$p(\mathbf{f}, \mathbf{f}_T | \mathbf{X}, \mathbf{X}_T) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_T \end{bmatrix}; \mathbf{0}, \mathbf{K}_{N+T} \right) \tag{4}$$

$$\mathbf{K}_{N+T} = \begin{bmatrix} \mathbf{K}_N & \mathbf{K}_{NT} \\ \mathbf{K}_{TN} & \mathbf{K}_T \end{bmatrix} \tag{5}$$

where $\mathbf{K}_T$ is a covariance matrix of test frames, and covariance matrix $\mathbf{K}_{NT} = \mathbf{K}_{TN}^\top$ consists of covariances between the training and test frames.

The joint distribution of $\mathbf{y}$ and $\mathbf{y}_T$ is given by

$$p(\mathbf{y}, \mathbf{y}_T | \mathbf{X}, \mathbf{X}_T) = \mathcal{N}\left(\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_T \end{bmatrix}; \mathbf{0}, \mathbf{K}_{N+T} + \sigma^2 \mathbf{I} \right). \tag{6}$$

Given a training data set, the predictive distribution of output variables of a test data set is obtained by

$$p(\mathbf{y}_T | \mathbf{y}, \mathbf{X}, \mathbf{X}_T) = \mathcal{N}\left(\mathbf{y}_T; \boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T\right) \tag{7}$$

$$\boldsymbol{\mu}_T = \mathbf{K}_{TN} [\mathbf{K}_N + \sigma^2 \mathbf{I}]^{-1} \mathbf{y} \tag{8}$$

$$\boldsymbol{\Sigma}_T = \mathbf{K}_T - \mathbf{K}_{TN} [\mathbf{K}_N + \sigma^2 \mathbf{I}]^{-1} \mathbf{K}_{NT}. \tag{9}$$

The inversion of $[\mathbf{K}_N + \sigma^2 \mathbf{I}]^{-1}$ requires $\mathcal{O}(N^3)$ computational cost[1]. For practical implementation, the parameter vector

$$\boldsymbol{\alpha} = [\mathbf{K}_N + \sigma^2 \mathbf{I}]^{-1} \mathbf{y} \tag{10}$$

that depends on the training data set only is calculated in the training phase. The number of parameters in $\boldsymbol{\alpha}$ is N, which corresponds to the number of frames of the training data. From (8), a set of new output means is given by an inner product

$$\boldsymbol{\mu}_T = \mathbf{K}_{TN} \boldsymbol{\alpha} \tag{11}$$

which requires $\mathcal{O}(NT)$ computational cost.

To use GPs for regression, we need to specify a kernel function. The necessary conditions for the kernel function are that the covariance matrix be positive semi-definite and symmetric. We use two typical kernels in this study: square exponential (SE) kernel and linear kernel. The SE kernel is one of the most widely used kernels as the measure of “similarity” between two input vectors. The SE kernel is defined by

$$k(\mathbf{x}_m, \mathbf{x}_n) = \exp\left(-\frac{\|\mathbf{x}_m - \mathbf{x}_n\|^2}{l^2} \right) \tag{12}$$

[1] To be exact, matrix inversion can be reduced from $\mathcal{O}(N^3)$. For instance, a Strassen algorithm enables $\mathcal{O}(N^{\log_2 7}) \approx \mathcal{O}(N^{2.807})$ computation complexity. However, note that the matrix inversion still requires very large computational cost.

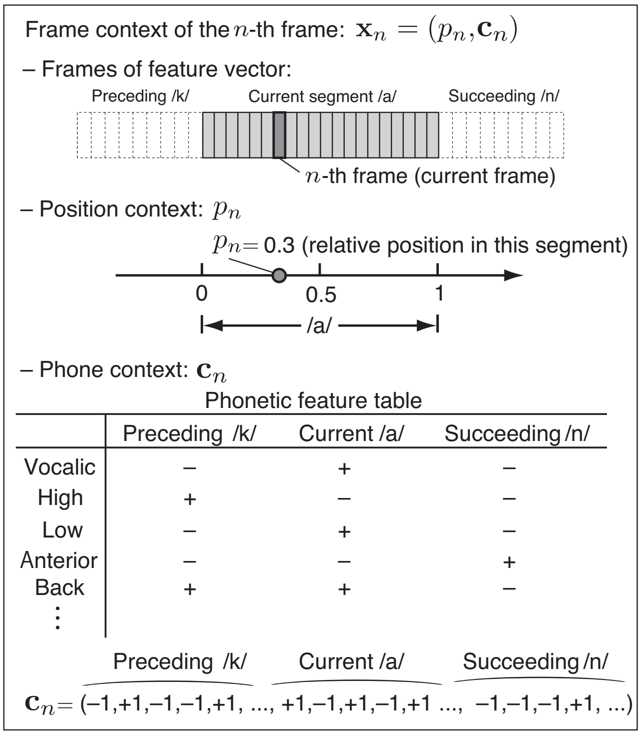

Frame context of the n-th frame: $\mathbf{x}_n = (p_n, \mathbf{c}_n)$
– Frames of feature vector:
Preceding /k/
Current segment /a/
Succeeding /n/
n-th frame (current frame)
– Position context: p_n
p_n= 0.3 (relative position in this segment)
0
0.5
1
/a/
– Phone context: $\mathbf{c}_n$
Phonetic feature table
Preceding /k/
Current /a/
Succeeding /n/
Vocalic – + –
High + – –
Low – + –
Anterior – – +
Back + + –
⋮
Preceding /k/
Current /a/
Succeeding /n/
$\mathbf{c}_n$= (–1,+1,–1,–1,+1, ..., +1,–1,+1,–1,+1 ..., –1,–1,–1,+1, ...)

Fig. 1. Example of frame context, i.e., frame-level input variable set for GP regression. This example has frame context for frame positioned in phone /a/, which is between preceding phone /k/ and succeeding phone /n/.

where l denotes a length-scale hyper-parameter. The linear kernel is given by

$$k(\mathbf{x}_m, \mathbf{x}_n) = \mathbf{x}_m^\top \mathbf{x}_n. \tag{13}$$

This kernel assumes linearity between output and input features. Also note that a new kernel can be constructed by combining multiple arbitrary kernel functions by means of some operations such as sum, product, and convolution [15].

B. Frame context with kernel design

We use frame-level features obtained from the linguistic information of transcriptions for the explanatory variables of the regression model. For the first step of an implementation of the proposed technique, we choose simple and compact representation of frame-level features. Specifically, we define the *frame context* that includes the relative position p_n and phonetic information $\mathbf{c}_n$ of the current frame as

$$\mathbf{x}_n = (p_n, \mathbf{c}_n). \tag{14}$$

Fig. 1 shows an example of the frame context. The position context p_n is defined as a normalized relative position [16]–[18] in the current phone, where the beginning of the phone is set to zero and its end is set to one. For phone context $\mathbf{c}_n$, we use a set of preceding, current, and succeeding phonetic features. More specifically, we introduce binary variables ($\{\text{positive} = +1, \text{negative} = -1\}$) for each phonetic feature listed in Table I based on a distinctive phonetic feature (DPF) set [19]. Let P be the number of phonetic features; then, a $3P$-dimensional binary-valued vector is constructed.

TABLE I
BINARY PHONETIC FEATURES FOR SEVERAL PRIMARY PHONEMES IN JAPANESE.

	a	i	u	e	o	k	t	n	s	m
vocalic	+	+	+	+	+	−	−	−	−	−
high	−	+	+	−	−	+	−	−	−	−
low	+	−	−	−	−	−	−	−	−	−
anterior	−	−	−	−	−	−	+	+	+	+
back	+	−	+	−	+	+	−	−	−	−
coronal	−	−	−	−	−	−	+	+	+	−
plosive	−	−	−	−	−	+	+	−	−	−
affricative	−	−	−	−	−	−	−	−	−	−
continuant	+	+	+	+	+	−	−	−	+	−
voiced	+	+	+	+	+	−	−	+	−	+
nasal	−	−	−	−	−	−	−	+	−	+
semi-vowel	−	−	−	−	−	−	−	−	−	−
silent	−	−	−	−	−	−	−	−	−	−

The proposed frame context kernel is defined as a product of two kernels.

$$k(\mathbf{x}_m, \mathbf{x}_n) = k_p(p_m, p_n) k_c(\mathbf{c}_m, \mathbf{c}_n) \tag{15}$$

where $k_p(p_m, p_n)$ and $k_c(\mathbf{c}_m, \mathbf{c}_n)$ correspond to the position kernel and the phone context kernel. The position kernel represents the similarity of position contexts in phones whereas the phone context kernel represents that of phone contexts.

1) Position kernel: The SE kernel is used for the position kernel and is given by

$$k_p(p_m, p_n) = \exp\left(-\frac{(p_m - p_n)^2}{l_p^2}\right) \tag{16}$$

where p_m is the relative position of the m-th frame.

2) Phone context kernel: We examine two different phone context kernels in this paper. The first is the sum of SE kernels, and the second is a linear kernel. The former one is defined by

$$k_c(\mathbf{c}_m, \mathbf{c}_n) = \sum_{k=1}^{3P} \theta_{ck}^2 \exp\left(-\frac{(c_{mk} - c_{nk})^2}{l_{ck}^2}\right) \tag{17}$$

where l_{ck} is a scale hyper-parameter, and θ_{ck} is a hyper-parameter that represents the relevance of the k-th phonetic feature. The kernel value is maximized when the input phone contexts are the same.

The linear kernel is given by

$$k_c(\mathbf{c}_m, \mathbf{c}_n) = \sum_{k=1}^{3P} \theta_{ck}^2 c_{mk} c_{nk}. \tag{18}$$

If we use only a linear kernel, GPR corresponds to a Bayesian inference of multiple linear regression. As a result, the use of this kernel implies that acoustic features in the same relative position can be modeled by the multiple linear regression.

C. GPR-based speech synthesis

Fig. 2 outlines a basic GPR-based speech synthesis system. When synthesizing speech, we generate a single feature se-

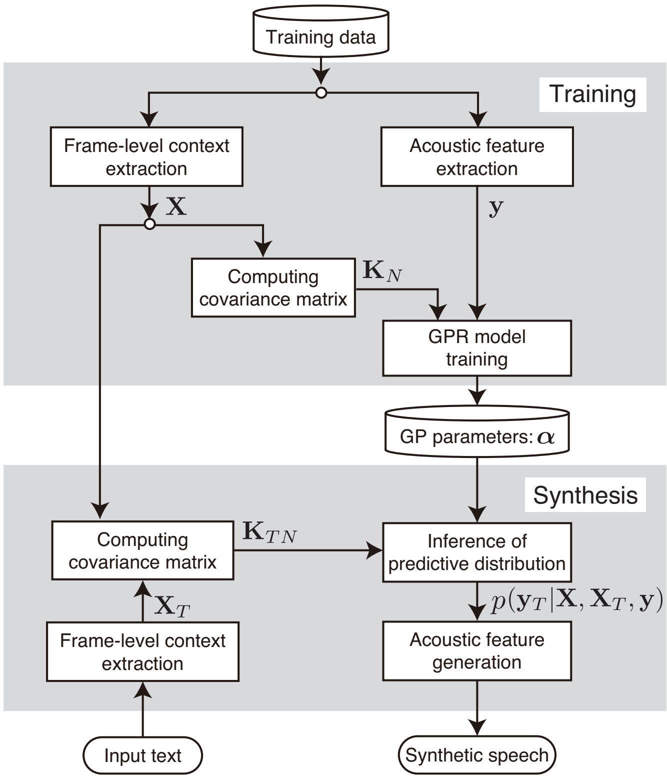

Training data
Training
Frame-level context extraction
Acoustic feature extraction
$\mathbf{X}$
$\mathbf{y}$
Computing covariance matrix
$\mathbf{K}_N$
GPR model training
GP parameters: $\boldsymbol{\alpha}$
Synthesis
Computing covariance matrix
$\mathbf{K}_{TN}$
Inference of predictive distribution
$p(\mathbf{y}_T|\mathbf{X}, \mathbf{X}_T, \mathbf{y})$
$\mathbf{X}_T$
Frame-level context extraction
Acoustic feature generation
Input text
Synthetic speech

Fig. 2. Outline of speech synthesis process in proposed approach.

quence from the predictive distribution with a certain method, such as using the mean sequence for synthetic parameters or generating random sequences from the distribution. In this study, we adopt the mean sequence, $\boldsymbol{\mu}_T$, of the predictive distribution.

Consequently, the training and synthesis procedures are summarized as follows:

Training phase

1) Frame-level acoustic features such as mel-cepstral coefficients and fundamental frequency are extracted from the training data.
2) The frame contexts are created from the annotation data including the phone boundaries of the training data.
3) Covariance matrix $\mathbf{K}_N$ between the frames of the training data is determined using the frame contexts.
4) Parameter vector $\boldsymbol{\alpha}$ in (10) is calculated using $\mathbf{K}_N$.

Synthesis phase

1) Phone duration information of an input text is predicted using a certain duration model.
2) The frame contexts are created from the input sentence and the predicted phone durations.
3) Covariance matrix $\mathbf{K}_{TN}$ between the frames of the training and new input data is calculated.
4) The mean sequence $\boldsymbol{\mu}_T$ of the predictive distribution is calculated by multiplying covariance matrix $\mathbf{K}_{TN}$ and $\boldsymbol{\alpha}$ and is used as a generated spectral feature trajectory.
5) The output waveform is synthesized using the spectral and excitation features.

III. Experiments on isolated phone synthesis

A. Experimental conditions

The speech database used in the experiments consisted of 503 ATR phonetically balanced Japanese sentences [20] spoken by one female speaker. Speech signals were sampled at a rate of 16 kHz. The spectral features were extracted with STRAIGHT [21]. The 0–39th mel-cepstral coefficients were used as output variables. We assumed that all dimensions of the acoustic features are conditionally independent given input context, and each dimension was modeled separately.

We chose five vowels (/a/, /i/, /u/, /e/, and /o/) and five consonants (/k/, /s/, /t/, /n/, and /m/), which are primary phonemes in Japanese, to examine the potential of GPR. Each phone was segmented using manually annotated phone boundaries. The phone segments of the training set were randomly chosen up to 10,000 frames from 450 sentences for each phoneme. Fifty test phone segments per phoneme were randomly chosen from the remaining 53 sentences. The evaluation was performed using the isolated phones segmented by the manually annotated phone boundaries of the original utterances. The durations of the original segments were given when the spectral feature sequences of test segments were generated.

We compared the sum of SE kernels and the linear kernel as the phone context kernels. All output variables were normalized, and the hyper-parameters were given by $l_p = 0.289$, which is equal to the standard deviation of the uniform distribution on $[0, 1]$, $l_{ck} = 1.0$ $(k = 1, \ldots, 3P)$, $\sigma = 1.0$, and $\theta_{ck} = 1.0/3P$ $(k = 1, \ldots, 3P)$ on the basis of the preliminary experimental results.

The HMM-based speech synthesis was used as a conventional technique. Triphones were used for the context set for HMM training. We used a five-state, left-to-right, no-skip hidden semi-Markov model (HSMM) [22]. Each state had a single Gaussian distribution with a diagonal covariance matrix and the feature vector included delta and delta-delta dynamic features. Decision-tree-based context clustering was carried out with the minimum description length (MDL) criterion [23]. State durations were generated using the trained HSMM. Minimum generation error (MGE) training [5] was not performed in this experiment.

B. Results

Table II lists the mel-cepstral distances [24] between the generated and original sequences. GPR-SE and GPR-linear employed the sum of SE kernels for the former and the linear kernel for the latter for the phone context kernel. Even though there were only small differences for consonants other than /s/ in comparing GPR with HMM, the mel-cepstral distances for the vowels using GPR-SE and GPR-linear significantly decreased. We also found that the distances for GPR-SE and GPR-linear were comparable. One possible reason is that the characteristics of kernel values were similar in the sum of SE kernels and the linear kernel under the condition in which the binary-valued vectors were used as input variables.

TABLE II
AVERAGE SPECTRAL DISTORTIONS OF GENERATED PARAMETER SEQUENCES USING FRAME CONTEXT. VALUES REPRESENT MEL-CEPSTRUM DISTANCES [dB].

Phoneme	triphone HMM	GPR-SE	GPR-linear
a	5.67	5.51	5.52
i	6.01	5.64	5.63
u	6.10	5.94	5.94
e	5.33	5.17	5.16
o	5.90	5.63	5.64
k	5.09	5.05	5.05
t	4.13	4.17	4.17
n	5.73	5.81	5.81
s	4.74	4.57	4.57
m	5.48	5.50	5.50
Avg.	5.42	5.30	5.30

IV. CONTINUOUS SPEECH SYNTHESIS BASED ON SPARSE GAUSSIAN PROCESSES

The matrix inversion needs $\mathcal{O}(N^3)$ calculations in the training procedure to obtain parameter $\boldsymbol{\alpha}$ in (10). The value of N is generally at least hundreds of thousands[2]. Therefore the computational complexity of GPR for continuous speech synthesis is not realistic. We examine two approximation methods to reduce the computational cost: local GPs [25], [26] and partially independent conditional (PIC) approximation [26]. These methods enable feasible computation by approximating matrices to be sparse. While various GP approximation methods exist [27], we chose the local GPs and PIC because they effectively model local characteristics within phone segments.

A. Local GPs

Using Local GPs involves a method for reducing the amount of computation by simply dividing all the data into local blocks and modeling each block separately. That is, covariance matrix $\mathbf{K}_{N+T}$ is approximated by a block diagonal one:

$$\begin{aligned}\mathbf{K}_{N+T} \approx \mathbf{K}_{N+T}^{\mathrm{LOCAL}} &= \mathrm{blkdiag}\left[\mathbf{K}_{N+T}\right] \\ &= \mathrm{diag}\left[\mathbf{K}_{B_1}, \mathbf{K}_{B_2}, \ldots, \mathbf{K}_{B_S}\right]. \qquad (19)\end{aligned}$$

When all training frames are divided into S blocks and each block has at most B training frames, the computational cost results in $\mathcal{O}(SB^3)$. By fixing B, the computational complexity increases linearly with the number of training data N.

To use the local GPs, it is necessary to determine not only the block of the training frames but also that of the synthesis frames from their linguistic features. We utilize decision-tree-based context clustering [4] in this study, which is effectively used in HMM-based speech modeling. We conduct phone-level clustering and stop splitting nodes if a node has less than B frames. After clustering, we use the clusters as the blocks, and compute the covariance matrix for each block using the frames included in the same cluster.

B. Partially independent conditional (PIC) approximation

Although the local GPs can model internally changing features effectively within a block, the covariances between different blocks are completely ignored. On the other hand, a partially independent conditional (PIC) approximation estimates the covariances between different blocks using a *pseudo-data set*. Pseudo-data set $\bar{\mathcal{D}} = \{(\bar{\mathbf{x}}_m, \bar{y}_m) | m = 1, \ldots, M\}$ is a small amount of data set with a size of $M \ll N$, and the pseudo-data are expected to be distributed similarly to the training data. PIC is a kind of approximation method called the sparse pseudo-input Gaussian process (SPGP) [28]. The joint distribution of the function values, $\mathbf{f}$ and $\mathbf{f}_T$, is given by a marginal distribution for pseudo-data variables $\bar{\mathbf{f}} = [f(\bar{\mathbf{x}}_1), \ldots, f(\bar{\mathbf{x}}_N)]^\top$ as

$$p(\mathbf{f}, \mathbf{f}_T) = \int p(\mathbf{f}, \mathbf{f}_T | \bar{\mathbf{f}}) p(\bar{\mathbf{f}}) d\bar{\mathbf{f}} \qquad (20)$$

where both $p(\mathbf{f}, \mathbf{f}_T | \bar{\mathbf{f}})$ and $p(\bar{\mathbf{f}})$ follow Gaussian distributions and are given by

$$p(\mathbf{f}, \mathbf{f}_T | \bar{\mathbf{f}}) = \mathcal{N}\left(\begin{bmatrix}\mathbf{f} \\ \mathbf{f}_T\end{bmatrix}; \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\Sigma}}\right) \qquad (21)$$

$$\bar{\boldsymbol{\mu}} = \mathbf{K}_{(N+T)M} \mathbf{K}_M^{-1} \bar{\mathbf{f}} \qquad (22)$$

$$\bar{\boldsymbol{\Sigma}} = \mathbf{K}_{N+T} - \mathbf{K}_{(N+T)M} \mathbf{K}_M^{-1} \mathbf{K}_{M(N+T)} \qquad (23)$$

$$p(\bar{\mathbf{f}}) = \mathcal{N}(\bar{\mathbf{f}}; \mathbf{0}, \mathbf{K}_M) \qquad (24)$$

where $\mathbf{K}_{(N+T)M}$ is a covariance matrix between the frames of all data $(\mathbf{X}, \mathbf{X}_T)$ and the pseudo-data, and $\mathbf{K}_M$ is a self covariance matrix of the pseudo-data set. SPGP is a method for avoiding the direct calculation of matrix inversion in (10) by approximating $p(\mathbf{f}, \mathbf{f}_T | \bar{\mathbf{f}})$. $\bar{\boldsymbol{\Sigma}}$ is approximated in PIC by using a block diagonal matrix as

$$\begin{aligned}\bar{\boldsymbol{\Sigma}} \approx \bar{\boldsymbol{\Sigma}}^{\mathrm{PIC}} &= \mathrm{blkdiag}[\bar{\boldsymbol{\Sigma}}] \\ &= \mathrm{diag}\left[\bar{\boldsymbol{\Sigma}}_{B_1}, \bar{\boldsymbol{\Sigma}}_{B_2}, \ldots, \bar{\boldsymbol{\Sigma}}_{B_S}\right]. \qquad (25)\end{aligned}$$

The covariance matrix of training data is approximated by

$$\mathbf{K}_N \approx \mathbf{K}_N^{\mathrm{PIC}} = \mathbf{Q}_N + \mathrm{blkdiag}\left[\mathbf{K}_N - \mathbf{Q}_N\right] \qquad (26)$$

$$= \begin{bmatrix} \mathbf{K}_{B_1} & \mathbf{Q}_{B_1 B_2} & \cdots & \mathbf{Q}_{B_1 B_S} \\ \mathbf{Q}_{B_2 B_1} & \mathbf{K}_{B_2} & & \mathbf{Q}_{B_2 B_S} \\ \vdots & & \ddots & \vdots \\ \mathbf{Q}_{B_S B_1} & \mathbf{Q}_{B_S B_2} & \cdots & \mathbf{K}_{B_S} \end{bmatrix} \qquad (27)$$

where $\mathbf{Q}_N$ and $\mathbf{Q}_{B_i B_j}$ are given by

$$\mathbf{Q}_N = \mathbf{K}_{NM} \mathbf{K}_M^{-1} \mathbf{K}_{MN} \qquad (28)$$

$$\mathbf{Q}_{B_i B_j} = \mathbf{K}_{B_i M} \mathbf{K}_M^{-1} \mathbf{K}_{M B_j}. \qquad (29)$$

$\mathbf{K}_{NM}$ and $\mathbf{K}_{MN}$ are covariance matrices whose elements are kernel values between the samples of training data and the pseudo-data set. Also, $\mathbf{K}_{B_i M}$ and $\mathbf{K}_{M B_j}$ are covariance matrices whose elements are kernel values between the samples of the clustered block and the pseudo-data set. Specifically, the approximation avoids direct calculations of inter-block covariance matrices by means of the pseudo-data set. Moreover, since $\mathbf{K}_N^{\mathrm{PIC}}$ is a sum of a block diagonal matrix and a low rank matrix $\mathbf{Q}_N$, we can speed up the inversion of $\mathbf{K}_N^{\mathrm{PIC}}$ using the Woodbury, Sherman & Morrison formula [29].

[2] If we have 10 min of speech data with 5-ms shift, N is 120,000.

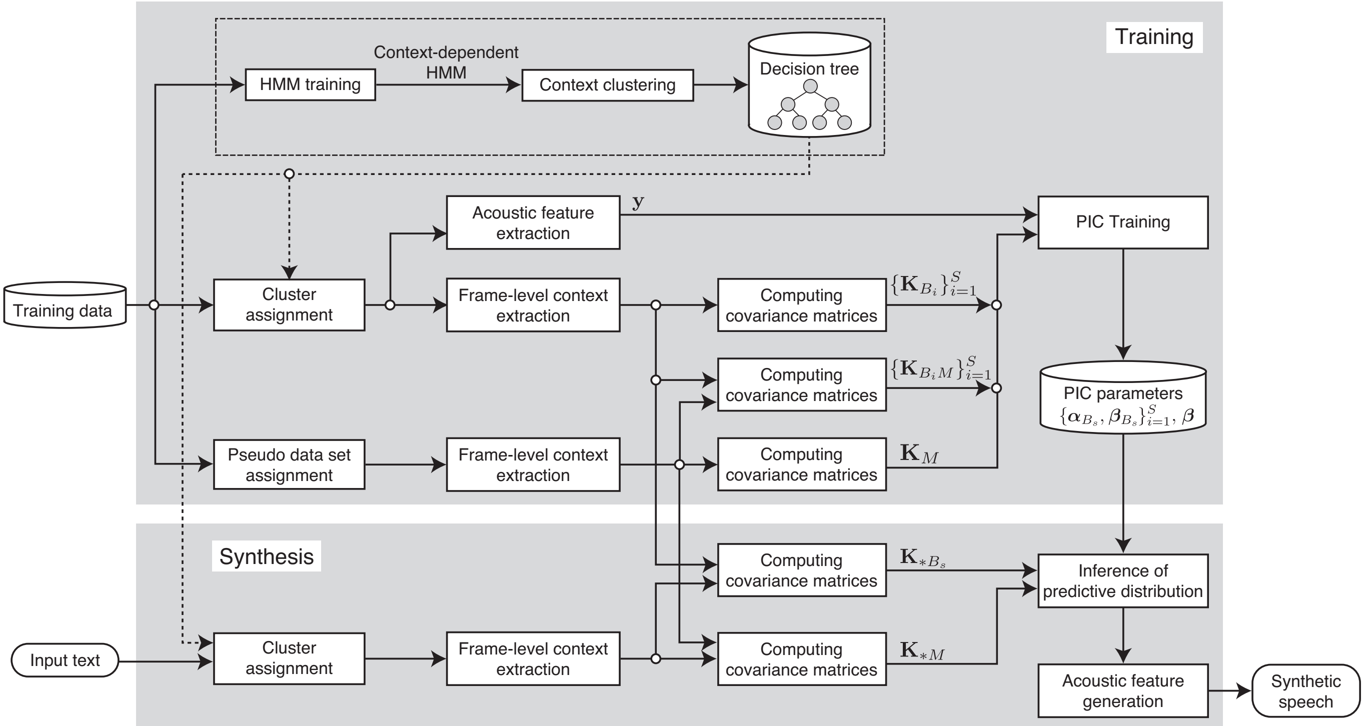

Training
Context-dependent HMM
HMM training
Context clustering
Decision tree
Acoustic feature extraction
y
PIC Training
Training data
Cluster assignment
Frame-level context extraction
Computing covariance matrices
$\{\mathbf{K}_{B_i}\}_{i=1}^{S}$
Computing covariance matrices
$\{\mathbf{K}_{B_iM}\}_{i=1}^{S}$
PIC parameters $\{\boldsymbol{\alpha}_{B_s}, \boldsymbol{\beta}_{B_s}\}_{i=1}^{S}, \boldsymbol{\beta}$
Pseudo data set assignment
Frame-level context extraction
Computing covariance matrices
$\mathbf{K}_M$
Synthesis
Computing covariance matrices
$\mathbf{K}_{*B_s}$
Inference of predictive distribution
Input text
Cluster assignment
Frame-level context extraction
Computing covariance matrices
$\mathbf{K}_{*M}$
Acoustic feature generation
Synthetic speech

Fig. 3. Overview of training and synthesis stages in GPR-based speech synthesis using PIC approximation.

When a new input value, $\mathbf{x}_*$, for a certain frame is assigned to cluster B_s, the corresponding mean for $\mathbf{x}_*$ is given by

$$\mu_* = \mathbf{K}_{*M}(\boldsymbol{\beta} - \boldsymbol{\beta}_{B_s}) + \mathbf{K}_{*B_s}\boldsymbol{\alpha}_{B_s} \tag{30}$$

where $\mathbf{K}_{*M}$ is a covariance matrix between the frames of $\mathbf{x}_*$ and the pseudo-data set, and $\mathbf{K}_{*B_s}$ is a covariance matrix between the frames of $\mathbf{x}_*$ and the s-th block data. The first and second terms on the right-hand side of (30) correspond to global and local acoustic characteristics. $\boldsymbol{\beta}$, $\boldsymbol{\beta}_{B_s}$, and $\boldsymbol{\alpha}_{B_s}$ are PIC model parameters calculated by

$$\boldsymbol{\beta} = \sum_{s=1}^{S} \boldsymbol{\beta}_{B_s} \tag{31}$$

$$\boldsymbol{\beta}_{B_s} = \mathbf{K}_M^{-1}\mathbf{K}_{MB_s}\boldsymbol{\alpha}_{B_s} \tag{32}$$

$$[\boldsymbol{\alpha}_{B_1}^\top \cdots \boldsymbol{\alpha}_{B_S}^\top]^\top = [\mathbf{K}_N^{\mathrm{PIC}} + \sigma^2\mathbf{I}]^{-1}\mathbf{y}. \tag{33}$$

When the maximum block size is B, the number of blocks is S, and the number of frames of the pseudo-data set is M, the computational cost results in $\mathcal{O}(S(B^3 + M^3))$. Here, the blocks of frames are determined in the same way as the local GPs. We adopt random selection from the training data to select the pseudo-data set.

Fig. 3 gives an overview of speech synthesis using PIC approximation. In the training phase, first, the decision tree of contexts is constructed using context-dependent HMMs. Then the pseudo-data set is chosen from the training data, and the cluster for each training data frame is assigned by the decision tree. Covariance matrices are computed after that. PIC parameters $\{\boldsymbol{\alpha}_i\}_{i=1}^S$, $\{\boldsymbol{\beta}_i\}_{i=1}^S$, and $\boldsymbol{\beta}$ in (31)–(33) are calculated at the end of the training phase. When speech is synthesized, the cluster for each frame context extracted from an input text is also determined by the decision tree. Next, covariance matrices between synthesis and training frames are computed and the acoustic features of the frames are generated from the covariance matrices and trained PIC parameters. Finally, a speech utterance is synthesized by using the generated spectral features.

C. Extension of frame context using adjacent phones

Even though PIC can express the covariances between different blocks, the simple frame context proposed in Section II-B is insufficient for synthesizing natural-sounding speech. A problem occurs when the simple frame context is used where covariances at the boundary of adjacent phones become discontinuous. For example, the context of the first frame of a current phone and that of the last frame of the preceding phone are entirely different. The discontinuity in covariance causes unsmoothness of the synthetic speech.

We extend the frame context to include smoothly changing values in order to overcome the discontinuity in covariance. Since a certain frame has information on not only the current phone but also nearby phones, extended frame context $\mathbf{x}_n$ is defined as a set of position and phone contexts of adjacent phones.

$$\mathbf{x}_n = (\mathbf{w}_n, \mathbf{p}_n, \mathbf{C}_n) \tag{34}$$

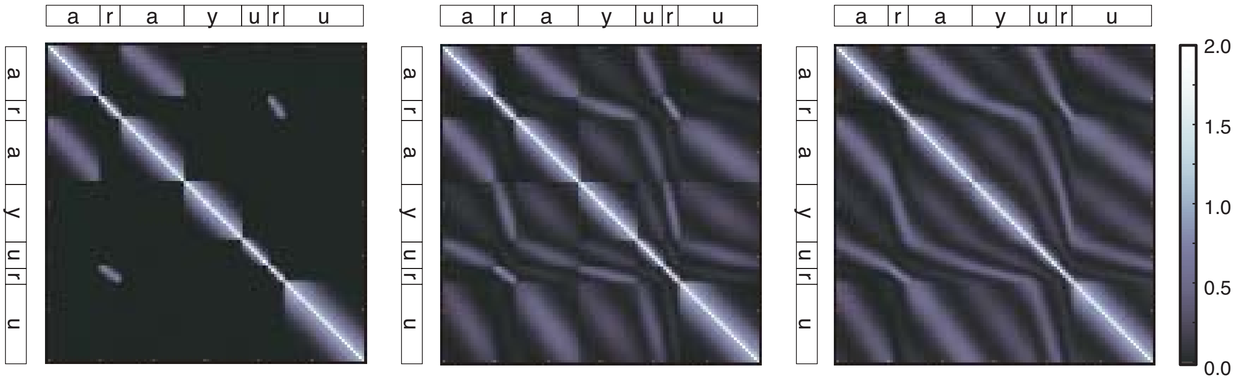
where $\mathbf{w}$, $\mathbf{p}$, and $\mathbf{C}$ are sets of weights, position contexts, and phone contexts expressed as

$$\mathbf{w}_n = \{w_n^{(-1)}, w_n^{(0)}, w_n^{(+1)}\} \tag{35}$$

$$\mathbf{p}_n = \{p_n^{(-1)}, p_n^{(0)}, p_n^{(+1)}\} \tag{36}$$

$$\mathbf{C}_n = \{\mathbf{c}_n^{(-1)}, \mathbf{c}_n^{(0)}, \mathbf{c}_n^{(+1)}\}. \tag{37}$$

The superscripts -1, 0, and $+1$ of the variables correspond to the preceding, current, and succeeding phones. Note that



(a) LOCAL-SIMPLE (b) PIC-SIMPLE (c) PIC-EXTEXDED

Fig. 4. Example of covariance matrices of Japanese phrase segment "a r a y u r u" using (a) local GPs and simple frame context, (b) PIC and simple frame context, and (c) PIC and extended frame contexts.

$p_n^{(0)}$ and $\mathbf{c}_n^{(0)}$ correspond to p_n and $\mathbf{c}_n$ in Section II-B. $p_n^{(-1)}$, $p_n^{(0)}$, and $p_n^{(+1)}$ respectively represent the normalized relative positions of the current frame in the preceding, current, and succeeding phones. $p_n^{(-1)}$ equals $p_n^{(0)} + 1$, and $p_n^{(+1)}$ equals $p_n^{(0)} - 1$. $\mathbf{c}_n^{(-1)}$, $\mathbf{c}_n^{(0)}$, and $\mathbf{c}_n^{(+1)}$ correspond to the phone context of preceding, current, and succeeding phones. $w_n^{(i)}$ represents the weight used to emphasize the effect of closer phones to the frame. For instance, the weight for the current phone is set to be larger than that for the preceding/succeeding phone. The following sine window function is used in this study as a weight.

$$w_n^{(i)} = w(p_n^{(i)}) = \begin{cases} \sin\left(\pi(p_n^{(i)} + 0.5)/2\right) & -0.5 \leq p_n^{(i)} \leq 1.5 \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

We use a convolution kernel [30], which computes the sum of all combinations between the adjacent phones of two input variables. The kernel function for the extended contexts is given by

$$k(\mathbf{x}_m, \mathbf{x}_n) = \sum_{i \in \{-1,0,+1\}} \sum_{j \in \{-1,0,+1\}} \left[w_m^{(i)} w_n^{(j)} k_p(p_m^{(i)}, p_n^{(j)}) k_c(\mathbf{c}_m^{(i)}, \mathbf{c}_n^{(j)}) \right]. \tag{39}$$

It is noted that the convolution kernel is positive semi-definite when each component kernel is positive semi-definite [30].

Fig. 4 shows examples of covariance matrices. Since the local GPs are used in Fig. 4 (a), many of the elements are zero because only intra-cluster covariances are defined. In contrast, inter-cluster covariances are estimated by using PIC in Fig. 4 (b) and (c). Moreover, the extended context in Fig. 4 (c) yields smooth covariances around the boundaries of adjacent phones.

D. Comparison with HMM-based synthesis

Table III compares memory footprint for model parameters and computational costs of HMM-based speech synthesis and GPR-based synthesis using local GPs and PIC. In the table, D represents the dimension of static feature vector. S' and L correspond to the number of leaf nodes in HMM-based speech synthesis and the dimensionality of dynamic features, e.g., $L = 2$ when delta and delta-delta features are used. It is assumed that HMM-based speech synthesis uses diagonal covariance matrices for Gaussian distribution.

The memory footprint of GPR is not compact because GPR is essentially a nonparametric model and requires as many parameters as training data, whereas HMM-based synthesis achieves compact representation. In the training of HMM-based synthesis, the Baum-Weltch algorithm computes the trellis of states and frames with $\mathcal{O}(NS'DL)$ complexity. Although the complexity is not small, required memory size is not large because of a sentence-by-sentence training. In contrast, covariance matrices among frames are computed in GPR training. If block size B or pseudo-data set size M becomes larger, it requires very large memory and computational time for inversion of the B-by-B or M-by-M matrix. For synthesis, both methods achieve the computational complexity of $\mathcal{O}(T)$.

V. Experiments on continuous speech synthesis

A. Experimental conditions

The speech database used in the experiments for continuous speech synthesis was the same as that used in Section III. Spectral envelope, F0, and aperiodicity features were extracted by using STRAIGHT [21]. The 0–39th mel-cepstral coefficients were normalized to zero means and used as output variables, and each dimension of the mel-cepstral coefficients was modeled separately, which was also the same condition as that in Section III. Speech samples were synthesized using generated mel-cepstral coefficients, while F0s, aperiodicity features, and phone durations were taken from the original speech.

The linear kernel was used for the position kernel. The hyper-parameters were set to $l_p = 0.289$, $\sigma = 1.0$, and $\theta_{ci} = 1.0/3P$ $(i = 1, \ldots, 3P)$, which were the same settings as those in Section III. $\mathbf{K}_M$ must be positive definite to calculate $\mathbf{K}_M^{-1}$ of (32) in PIC, and hence the value of $\theta_\delta \delta_{mn}$ was added to the kernel function $k(\mathbf{x}_m, \mathbf{x}_n)$ where δ_{mn} is the Kronecker delta. The value of θ_δ was set to unity on the basis of preliminary experimental results. For context clustering in the local GPs and PIC, the model topology of HMM and feature vector including dynamic features were the same as those in the case of the HMM-based method in Section III.

TABLE III
COMPARISON OF COMPUTATIONAL COMPLEXITY OF GPR-BASED TECHNIQUES WITH HMM-BASED SPEECH SYNTHESIS.

	HMM	GPR (local GPs)	GPR (PIC)
Memory footprint for model parameters	$\mathcal{O}(S'DL)$	$\mathcal{O}(SDB)$	$\mathcal{O}(SD(B+M))$
Memory for training	$\mathcal{O}(S'DL)$	$\mathcal{O}(SDB^2)$	$\mathcal{O}(SD(B^2+MB+M^2))$
Complexity for training	$\mathcal{O}(NS'DL)$	$\mathcal{O}(SDB^3)$	$\mathcal{O}(SD(B^3+M^3)$
Memory for synthesis	$\mathcal{O}(TDL)$	$\mathcal{O}(TDB)$	$\mathcal{O}(TD(B+M))$
Complexity for synthesis	$\mathcal{O}(TDL^2)$	$\mathcal{O}(TDB)$	$\mathcal{O}(TD(B+M))$

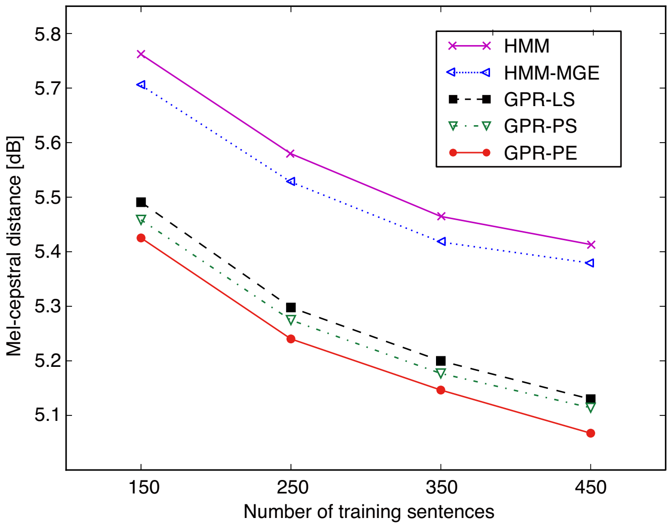


Fig. 5. Average spectral distortions between original and synthetic speech as a function of the number of training sentences.

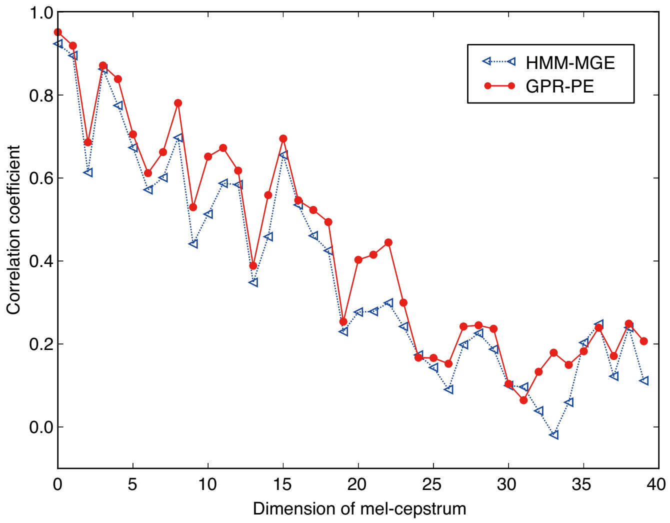


Fig. 6. Correlation coefficients between original and generated mel-cepstral coefficients.

Triphones were used for the context set for HMM training. The maximum number of frames, B, of the cluster described in Section IV-B was set to 1000, and the size of pseudo-data set M was set to 200.

We also evaluated HMM-based speech synthesis with and without minimum generation error (MGE) training [5] for comparison. The model topology of HMM, feature vector, and context set were the same as those used in the context clustering in the local GPs and PIC. The MDL was used for a stopping criterion for context clustering.

In subjective evaluation, test volunteers were ten native Japanese speakers, who were university students and researchers. Participants listened to the test samples using a pair of headphones in a silent room[3].

B. Objective evaluation

First, we objectively compared the performance of the conventional and proposed techniques. The mel-cepstral distance between synthetic and original speech were used as an objective distortion measure. We used 150, 250, 350, and 450 sentences as the training data, and 53 sentences not included in the training data were used as the test data. We compared two kinds of HMM-based techniques and three kinds of proposed GPR-based techniques. The results are plotted in Fig. 5. "HMM" in the figure represents the HMM-based technique where the model parameters were optimized by the maximum likelihood (ML) criterion. "HMM-MGE" used MGE training to optimize the model parameters. In the proposed GPR-based techniques, L and P corresponded to local GPs and PIC for approximation, and S and E denote the simple frame context and the extended frame context. We can see that both HMM-MGE and GPR-based techniques gave smaller distortions than HMM. Moreover, the GPR-based techniques derived significantly smaller distortions than HMM-MGE, which means that frame-level regression performed well. We could see that distortions decreased slightly for all the training sets by comparing GPR-LS and GPR-PS. In addition, GPR-PE had consistently less distortion than GPR-PS.

Next, we compared the HMM-MGE and proposed GPR-PE techniques in terms of the correlation between the original and generated mel-cepstral coefficients. In this experiment, models trained on 450 sentences were used, and correlation coefficients were calculated for each dimension. From the results shown in Fig. 6, we can see that the correlation coefficients of the proposed GPR-based technique were higher than those of the HMM-based technique in most dimensions. This improvement may be attributed to the advantages of GPR such as nonparametric modeling, context-space representation by kernel function, and frame-level inference without using dynamic features.

Computational time was evaluated to examine whether the realization of GPR-PE is feasible or not. We used a 64bit

[3] Some examples of the synthetic speech used in the subjective evaluation are available at http://www.kbys.ip.titech.ac.jp/demo/gpss/koriyama/

TABLE IV
AVERAGE COMPUTATION TIME FOR MODEL TRAINING AND PARAMETER GENERATION. THE GENERATION TIME REPRESENTS THE AVERAGE TIME FOR TEST 53 SENTENCES.

Training data size		Training	Generation
50 sentences	(282.1 sec)	326.8 sec	2.82 sec
150 sentences	(886.2 sec)	1039.2 sec	3.48 sec
250 sentences	(1436.1 sec)	1731.3 sec	3.67 sec
350 sentences	(1968.6 sec)	2448.9 sec	4.22 sec
450 sentences	(2492.9 sec)	3023.0 sec	4.32 sec

FreeBSD 8.3 PC with Intel Core i7 3770K 3.50GHz and 32GB/PC3-12800 memory. Table IV shows actual computational time for training with PIC approximation and acoustic feature generation. Each value was an average of ten trials. The training time consists of computing of covariance matrices and training of PIC parameters. The generation time represents the average time of covariance matrix computation and acoustic feature inference for 53 test sentences. The results show that the training time was approximately 120% as much as the time of training utterances. The generation time for 1 sentence was shorter than 5 seconds. This implies that a slight effort is needed to achieve real time synthesis because the average length of a synthetic utterance was 4.11 seconds.

C. Subjective evaluation

1) Naturalness: We evaluated HMM-MGE, GPR-LS, and GPR-PE by using a mean opinion score (MOS) test to subjectively examine the naturalness of the synthetic speech samples. There were 450 training sentences. The listeners rated the naturalness of synthetic speech on a five-point scale: 5: excellent, 4: good, 3: fair, 2: poor, and 1: bad. Ten sentences were randomly chosen from the 53 sentences for each participant. Fig. 7 shows the mean opinion scores (MOSs). The error bars indicate 95% confidence intervals based on a t-distribution. We can see that the score of GPR-LS was lower than that of HMM-MGE, whereas GPR-LS derived smaller mel-cepstral distances in the objective evaluation. This is because the generated acoustic features were not smooth at the phone boundaries and this discontinuity degraded naturalness. In contrast, GPR-PE, which provided continuity on covariance matrices, gave the highest score for the three techniques although GPR-PE and HMM-MGE do not differ significantly at the 5% significance level according to a t-test. When we carefully listened to the synthetic speech samples of the proposed technique, some plosives, e.g., /k/ and /t/, sounded unnatural. This can be observed in the spectral sequence of synthetic utterance shown in Fig. 8. In the spectra of original speech, we can see a "stop gap" which corresponds to the period of closure of vocal tract. However we cannot see such gap in the generated spectrum of GPR-PE. This is one of the reasons why plosives sounded unnatural.

2) Similarity: We conducted an XAB test on speech similarity between vocoded and synthetic speech samples to compare the reproducibility of synthetic speech samples with the conventional and proposed techniques. Ten sentences were randomly chosen from the 53 test sentences for each of the participants. After being given a vocoded speech sample (X) as a reference, the participants listened to two synthetic speech samples (A and B) in random order and were asked whether A or B was closer to X. We used synthetic speech samples with all combinations of the three techniques for the pairs of A and B. Fig. 9 shows the results of XAB test with 95% confidence intervals assuming a binomial distribution. Although there are no statistically significant differences among the scores, we can see that GPR-PE gives slightly higher score than HMM-MGE.

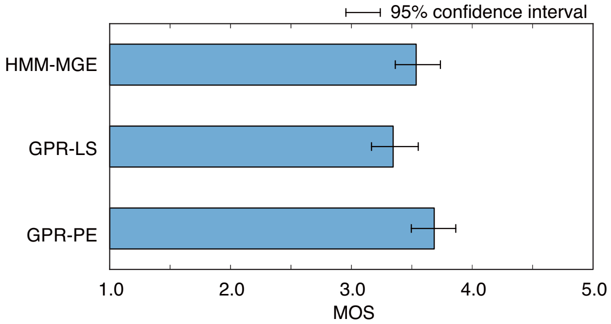
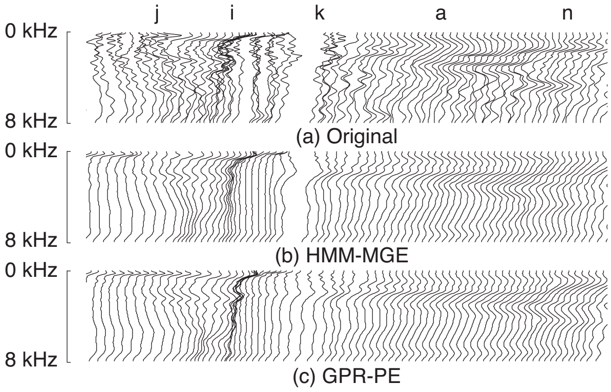


Fig. 7. MOS on naturalness of synthetic speech.



(a) Original

(b) HMM-MGE

(c) GPR-PE

Fig. 8. Example of running spectra of a Japanese word "jikan" ("time" in English). (a) extracted from original speech, (b) generated using HMM-MGE, and (c) generated using GPR-PE.

VI. DISCUSSIONS

A. TTS system

In this study, we focused only on the modeling and synthesis of spectral features. To achieve a full TTS system, we must examine the same issue for the other features, i.e., F0, duration, and aperiodicity features. For this purpose, it is essential to take into account not only the phonetic contexts but also the prosodic ones, e.g., syllable, accent phrase, and sentence length that are usually used in HMM-based speech synthesis. It will be also important to examine the choice of appropriate speech synthesis unit other than the phone-based segment, e.g., diphone-based segment.

B. GP approximation

In the experiments, the maximum number of frames of each cluster, B, was set to 1000 and the size of pseudo-

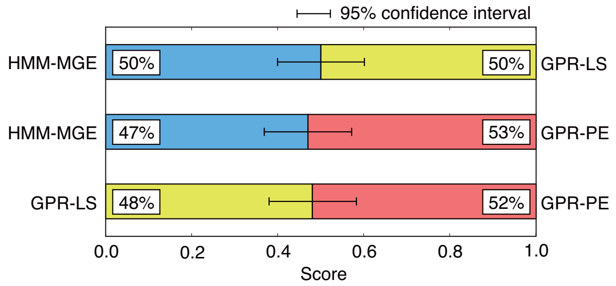


Fig. 9. XAB test score on similarity of synthetic speech to original speech.

data set M was set to 200. Increasing these parameters may improve the accuracy because approximated GPs become closer to full (non-approximated) GPs. Since computational cost increases with these size parameters, the relationship between the performance and computational cost should be examined for a practical TTS system. The selection of a pseudo-data set, which was randomly chosen in this study, should be also investigated. For example, using the decision-tree yields a balanced pseudo-data set and may lead to more effective modeling.

C. Parameter generation

Using the predictive mean sequence generates the most likely sequence but generally causes over-smoothing. An alternative way is incorporating a global variance (GV) [31] or postfiltering [32], [33]. The GV model works as a restriction in parameter generation. Another way is sampling from the predictive distribution. It is reported that the sampled trajectories using HMM-based methods sounds very artificial and unnatural [34]. However, the explicit covariance representation of frames in GPs may produce more appropriate trajectories.

D. Kernel design

We need to adjust hyperparameters of kernels, which were manually tuned on the basis of preliminary experimental results in this study. To optimize hyperparameters such as scales and relevance weights, we can use Type-II maximum likelihood, which maximizes the marginal likelihood of all training frames [14]. Kernel selection is another important issue. Instead of choosing either a linear or SE kernel, we can use a combination of them. We can also examine other kernels. For instance, a neural network kernel was used effectively for terrain data [35] and represents sharp feature variation like cliffs, whereas SE kernel generally assumes smoothly changing features. The neural network kernel could overcome the problem of trajectories that are too smooth in plosive phonemes. Furthermore, the window function for the convolution kernel should be examined.

E. Input variables

The input features used in this paper were quite simple and compact. The linearly normalized position loses the phone characteristics that have rapid changes at the beginning and end of the phone. Therefore, we need to consider additional features for position context using not only linear normalization but also other techniques, e.g., warping function for normalization [36] and alignment of sub-phones such as short silence before plosive. For example, since the normalized position ignores the distinction of phone duration, unnormalized position should be added to input features. In addition, phone context can be extended by using questions for individual phones, such as whether the phoneme is /i/ or not.

VII. Conclusions

This paper proposed a novel approach to speech synthesis using Gaussian process regression (GPR). We first described the basic framework of GPR-based speech synthesis and evaluated it using a small data set of isolated phones. We then achieved continuous speech synthesis with feasible computational cost using partially independent conditional (PIC) approximation and context extension. The evaluation results revealed that introducing PIC and context extension into the proposed technique effectively reduced spectral distortion. However, the naturalness of synthetic speech was comparable with HMM-based technique and there are still many issues to using GPR-based speech synthesis as a general and useful system. In addition, the current study focused only on the spectral feature modeling and had a lack of prosody modeling. In our future work, we will examine the modeling F0, duration, and aperiodicity features using both phonetic and prosodic contexts. It is also important to compare the performances of GPR- and HMM-based speech synthesis under the better conditions where the hyperparameters and/or model complexity are manually/automatically tuned and controlled.

References

[1] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis,” in *Proc. EUROSPEECH*, 1999, pp. 2347–2350.

[2] H. Zen, K. Tokuda, and A. Black, “Statistical parametric speech synthesis,” *Speech Communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] K. Tokuda, T. Kobayashi, and S. Imai, “Speech parameter generation from HMM using dynamic features,” in *Proc. ICASSP-95*, 1995, pp. 660–663.

[4] J. J. Odell, “The use of context in large vocabulary speech recognition,” Ph.D. dissertation, University of Cambridge, 1995.

[5] Y. J. Wu and R. H. Wang, “Minimum generation error training for HMM-based speech synthesis,” in *Proc. ICASSP*, vol. 1, 2006, pp. 889C–892.

[6] H. Zen, K. Tokuda, and T. Kitamura, “Reformulating the HMM as a trajectory model by imposing explicit relationships between static and dynamic feature vector sequences,” *Computer Speech & Language*, vol. 21, no. 1, pp. 153–173, 2007.

[7] M. Shannon and W. Byrne, “Autoregressive HMMs for speech synthesis,” in *Proc. Interspeech*, vol. 2009, 2009, pp. 400–403.

[8] J. Yu, M. Zhang, J. Tao, and X. Wang, “A novel HMM-based TTS system using both continuous HMMs and discrete HMMs,” in *Proc. ICASSP*, 2007, pp. 709–712.

[9] Z.-J. Yan, Y. Qian, and F. K. Soong, “Rich context modeling for high quality HMM-based TTS,” in *Proc. INTERSPEECH*, 2009, pp. 1755–1758.

[10] S. Park and S. Choi, “Gaussian process regression for voice activity detection and speech enhancement,” in *Proc. IJCNN*, 2008, pp. 2879–2882.

[11] N. C. V. Pilkington, H. Zen, and M. J. F. Gales, “Gaussian process experts for voice conversion,” in *Proc. INTERSPEECH*, 2011, pp. 2761–2764.

[12] H. Park and C. D. Yoo, “Gaussian process dynamical models for phoneme classification,” in *NIPS 2011 Workshop on Bayesian Nonparametrics: Hope or Hype*, 2011.

[13] G. Henter, M. Frean, and W. Kleijn, “Gaussian process dynamical models for nonparametric speech representation and synthesis,” in *Proc. ICASSP*, 2012, pp. 4505–4508.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT press Cambridge, MA, 2006.

[15] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[16] R. Terashima, H. Zen, Y. Nankaku, and K. Tokuda, “A frame-based context-dependent acoustic modeling for speech recognition (Japanese),” *IEEJ Transactions on Electronics, Information and Systems*, vol. 130, no. 10, pp. 1856–1864, 2010.

[17] T. Koriyama, T. Nose, and T. Kobayashi, “An F0 modeling technique based on prosodic events for spontaneous speech synthesis,” in *Proc. ICASSP*, 2012, pp. 4589–4593.

[18] H. Zen, A. Senior, and M. Schuster, “Statistical parametric speech synthesis using deep neural networks,” in *Proc. ICASSP*, 2013, pp. 7962–7966.

[19] T. Fukuda and T. Nitta, “Orthogonalized distinctive phonetic feature extraction for noise-robust automatic speech recognition,” *IEICE Trans. Inf. & Syst.*, vol. 87, no. 5, pp. 1110–1118, 2004.

[20] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K. Shikano, “ATR Japanese speech database as a tool of speech recognition and synthesis,” *Speech Communication*, vol. 9, no. 4, pp. 357–363, Aug. 1990.

[21] H. Kawahara, I. Masuda-Katsuse, and A. de Cheveigne, “Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds,” *Speech Communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[22] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “A hidden semi-Markov model-based speech synthesis,” *IEICE Trans. Inf. & Syst.*, vol. 90, no. 5, pp. 825–834, 2007.

[23] K. Shinoda and T. Watanabe, “MDL-based context-dependent subword modeling for speech recognition,” *Acoustical Science and Technology*, vol. 21, no. 2, pp. 79–86, 2000.

[24] R. Kubichek, “Mel-cepstral distance measure for objective speech quality assessment,” in *Communications, Computers and Signal Processing, 1993., IEEE Pacific Rim Conference on*, vol. 1, 1993, pp. 125–128.

[25] H. Wackernagel, *Multivariate Geostatistics*. Springer, 2003.

[26] E. Snelson and Z. Ghahramani, “Local and global sparse Gaussian process approximations,” in *Proc. AISTATS*, 2007.

[27] J. Quiñonero-Candela and C. E. Rasmussen, “A unifying view of sparse approximate Gaussian process regression,” *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[28] E. Snelson and Z. Ghahramani, “Sparse Gaussian processes using pseudo-inputs,” in *In NIPS 18, MIT press*, 2006, pp. 1257C–1264.

[29] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge Univ. Press, 1992.

[30] D. Haussler, “Convolution kernels on discrete structures,” in *Technical Report UCSC-CRL-99-10*. Dept of Computer Science, University of California at Santa Cruz., 1999.

[31] T. Toda and K. Tokuda, “A speech parameter generation algorithm considering global variance for HMM-based speech synthesis,” *IEICE Trans. Inf. & Syst.*, vol. E90-D, no. 5, pp. 816–824, 2007.

[32] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Mixed excitation for HMM-based speech synthesis,” in *Proc. Eurospeech*, 2001, pp. 2263–2266.

[33] Z.-H. Ling, Y.-J. Wu, Y.-P. Wang, L. Qin, and R.-H. Wang, “USTC system for Blizzard Challenge 2006 an improved HMM-based speech synthesis method,” in *Blizzard Challenge Workshop*, 2006.

[34] M. Shannon, H. Zen, and W. Byrne, “The effect of using normalized models in statistical speech synthesis,” in *Proc. Interspeech*, vol. 2011, 2011, pp. 121–124.

[35] S. Vasudevan, F. Ramos, E. Nettleton, and H. Durrant-Whyte, “Nonstationary dependent Gaussian processes for data fusion in large-scale terrain modeling,” in *Proc. ICRA*, 2011, pp. 1875–1882.

[36] E. Snelson, C. E. Rasmussen, and Z. Ghahramani, “Warped Gaussian processes,” in *In NIPS 16, MIT press*, 2004, pp. 337–344.

Tomoki Koriyama (M’13) received the B.E. degree in computer science, the M.E. and Ph.D degrees in information processing from Tokyo Institute of Technology, Tokyo, Japan, in 2009, 2010, and 2013, respectively.

He is currently a Research Fellow of Japan Society for the Promotion of Science (JSPS) in the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan. His research interests include speech synthesis, spoken language processing, and human computer interaction. He is a member of IEEE, ISCA, IEICE, and ASJ.

Takashi Nose (M’10) received the B.E. degree in electronic information processing, from Kyoto Institute of Technology, Kyoto, Japan, in 2001. He received the Dr.Eng. degree in information processing from Tokyo Institute of Technology, Tokyo, Japan, in 2009.

He was a Ph.D. researcher of The 21st Century Center Of Excellence (COE) program and Global COE program in 2006 and 2007, respectively. He was an Intern Researcher at ATR spoken language communication Research Laboratories (ATR-SLC) from July 2008 to January 2009.He is currently an Assistant Professor of the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan. His research interests include speech synthesis, speech recognition, and voice conversion.

He is a member of IEEE, ISCA, IEICE, and ASJ.

Takao Kobayashi (M’82-SM’04) received the B.E. degree in electrical engineering, the M.E. and Dr.Eng. degrees in information processing from Tokyo Institute of Technology, Tokyo, Japan, in 1977, 1979, and 1982, respectively.

In 1982, he joined the Research Laboratory of Precision Machinery and Electronics, Tokyo Institute of Technology as a Research Associate. He became an Associate Professor at the same Laboratory in 1989. He is currently a Professor of the Interdisciplinary Graduate School of Science and Engineering, Tokyo Institute of Technology, Yokohama, Japan.

Dr. Kobayashi was a corecipient of both the Best Paper Award and the Inose Award from the IEICE in 2001, and the TELECOM System Technology Prize from the Telecommunications Advancement Foundation Award, Japan, in 2001 and 2008. He was also a recipient of IEICE Information and Systems Society Distinguished Service Award in 2010. He served as the chair of the Speech Committee of IEICE and ASJ in 2007 and 2008. He is a fellow of IEICE, and a member of IEEE, ISCA, ASJ and IPSJ.