International Journal of Multimedia and Ubiquitous Engineering
Vol.9, No.5 (2014), pp.107-120
http://dx.doi.org/10.14257/ijmue.2014.9.5.10

HMM-Based Distributed Text-to-Speech Synthesis Incorporating Speaker-Adaptive Training

Kwang Myung Jeon[1] and Seung Ho Choi[2,*]

[1]School of Information and Communications
Gwangju Institute of Science and Technology, Gwangju 500-712, Korea
[2] Dept. of Electronic and IT Media Engineering
Seoul National University of Science and Technology, Seoul 139-743, Korea
E-mail: shchoi@seoultech.ac.kr
** Corresponding author*

Abstract

In this paper, a hidden Markov model (HMM) based distributed text-to-speech (TTS) system is proposed to synthesize the voices of various speakers in a client-server framework. The proposed system is based on speaker-adaptive training for constructing HMMs corresponding to a target speaker, and its computational complexity is balanced by distributing the processing modules of the TTS system at both the client and server to achieve a real-time operation. In other words, fewer complex operations, such as text inputs and HMM-based speech synthesis, are conducted by the client, while speaker-adaptive training, which is a very complex operation, is assigned to the server. It is shown from performance evaluation that the proposed TTS system operates in real time and provides good synthesized speech quality in terms of intelligibility and similarity.

Keywords: *Text-to-speech (TTS), hidden Markov model, distributed processing, client–server processing, speaker-adaptive training, harmonic modeling, non-harmonic modeling*

1. Introduction

Speech signal processing techniques have been developed in various ways such as signal enhancement [1-5], signal transmission over wireless network [6], speech recognition, and synthesis [7-8] to meet demanding of numerous speech-based applications. Among these techniques, speech synthesis is one of the important tasks because most applications including speech-based user interface requires various kinds of voices to be generated by a single text-to-speech (TTS) system [8]. To meet such a demand, commercially available TTS systems based on a concatenative method have applied a set of large-scale speech databases with various types of voices [9]. However, such an approach is both time- and cost-consuming for expressing various types of voices since it requires a large set of speech data recorded by a target speaker to add a new speaking style. This is also because the performance of a concatenative TTS system depends highly on the database size [8]. In contrast to such a conventional concatenative TTS method, a hidden Markov model (HMM) based TTS method with speaker-adaptive training was proposed [10]. Compared to the concatenative TTS method, the HMM-based TTS method with speaker-adaptive training can imitate a voice of the target speaker using a considerably small amount of recorded speech data of the target-speaker. In particular, adapted HMMs for the target speaker have been obtained from average voice models created using hidden semi-Markov models (HSMMs) or model adaptation

ISSN: 1975-0080 IJMUE

techniques [10-12]. However, since the speaker-adaptive training of an HMM-based TTS method requires a higher computational power, it may be impractical to implement the HMM-based TTS method with speaker-adaptive training in an embedded system with limited computational resources.

In this paper, we propose an HMM-based distributed TTS system in a client-server framework. In other words, lower complex operations, such as text inputting and speech synthesis, are performed at the client side, while speaker-adaptive training is performed on the server since they are computationally burdensome to most embedded systems. By distributing the processing modules of the proposed TTS system at both the client and the server, the TTS system can operate in real time.

Following this introduction, Section 2 discusses the issues associated with HMM-based TTS. Section 3 then describes the proposed HMM-based distributed TTS system, and discusses speaker-adaptive training in detail. In Section 4, the performance of the proposed TTS system is evaluated in terms of the processing time required for the speaker-adaptive training process and the quality of the synthesized speech. Finally, Section 5 concludes this paper.

2. HMM-Based TTS System with Speaker-Adaptive Training

An HMM-based TTS system tries to generate various voices without relying on a large speech database by using speaker adaptation techniques [10]. Such a system is typically composed of training and synthesis parts. In the training part, average voice models are first created using speech data from several speakers, where they are represented as context-dependent HMMs that are trained using feature vectors consisting of excitation and spectral parameters. In addition, the temporal structure of speech is modeled using a state duration probability density function for each HMM. Finally, the average voice models are adapted using a small amount of speech data recorded by the target speaker. Several speaker adaptation techniques have been used for speaker-adaptive training, including constrained maximum likelihood linear regression (CMLLR) [11], structural maximum a posteriori linear regression (SMAPLR) [12], and constrained SMAPLR (CSMAPLR) [13].

In the synthesis part, speech signals are synthesized using the HMMs from a given input text. In other words, the input text is analyzed and labeled using information on the language-specific contexts, which could be phonetic-, segment-, syllable-, word-, and utterance-level features [14]. An utterance HMM is then constructed by concatenating the context-dependent HMMs corresponding to the label sequence. Next, the sequence of spectral and excitation parameters is generated from the utterance HMM using a parameter generation algorithm [15]. Finally, synthesized speech signals are obtained using the estimated excitation and spectral parameters.

As mentioned earlier, speaker-adaptive training plays a main role in synthesizing speaker-specific voices, which is a process requiring an excessive amount of computational complexity. If an HMM-based TTS system is to be implemented on re-source-constrained devices with network connections such as mobile phones and smart TVs, it is necessary to reduce the complexity of speaker-adaptive training. Instead of directly reducing this complexity, a distributed approach can be an alternative. In other words, speaker-adaptive training is performed on a server, while relatively low complexity operations, such as feature extraction and speech synthesis, are performed at the client side. Such a distribution approach enables a TTS system with speaker-adaptive training to work in real time.

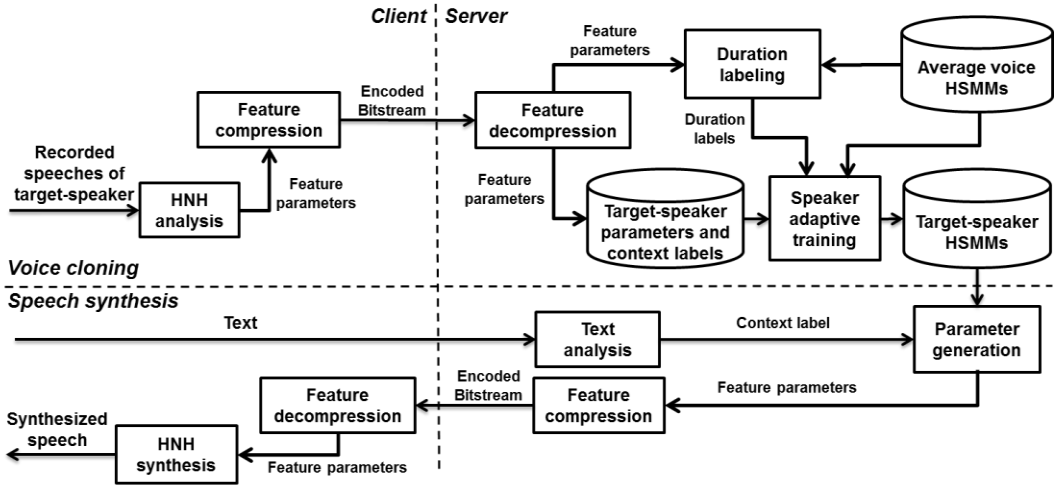


Figure 1. Block Diagram of the Proposed HMM-based Distributed TTS System

3. Proposed HMM-Based Distributed TTS System

Figure 1 shows a block diagram of the proposed HMM-based distributed TTS system. As shown in the figure, the proposed system is divided into two stages: voice cloning and speech synthesis. All server and client operations for the voice cloning and speech synthesis stages are described in the following subsections.

3.1. Voice Cloning Stage

In this stage, the voice of the target speaker is recorded at the client side. It was shown from the results of a preliminary experiment that the amount of target speech sufficient for cloning was around 10 min in length. However, the target speaker should pronounce phonetically balanced sentences to achieve efficient speaker-adaptive training. Therefore, 100 phonetically balanced sentences are selected from the CMU US ARCTIC speech database [16] for the speech data of the target speaker.

3.1.1. Harmonic and Non-Harmonic Modeling of Speech: After the recording of the target speeches, a set of feature parameters is extracted once every frame for the adaptation. In this paper, harmonic and non-harmonic (HNH) modeling [17] is applied for speech analysis and synthesis. First, each utterance of the target speaker is segmented into a sequence of frames, where the frame length, *N*, is set to 512 samples. In addition, each frame is overlapped with adjacent frames by as many as (*N* – *L*) samples, where *L* is the frame shift length and is set to 40 samples. Thus, N and L correspond to 32 ms and 2.5 ms, respectively, at a sample frequency of f_s = 16 kHz. Note that N and L are chosen to analyze the harmonic component with a fundamental frequency ranging from 31.25 Hz to 400 Hz within a frame.

To obtain the parameters for the HNH modeling, it is assumed that the speech signal of the *l*-th frame, $s^{(l)}(n)$, is composed of a harmonic component, $s_h^{(l)}(n)$, and a non-harmonic component, $s_{nh}^{(l)}(n)$, as

$$s^{(l)}(n) = s_h^{(l)}(n) + s_{nh}^{(l)}(n) \quad (1)$$

The decomposition of (1) is realized depending on the degree of harmonicity of the l-th frame. Initially, a robust pitch-tracking algorithm is applied to $s^{(l)}(n)$ in order to obtain the fundamental frequency, $F_0^{(l)}$, where a robust algorithm for pitch tracking (RAPT) [18] is used. Note that $F_0^{(l)}$ is set to 0 if the *l*-th frame has little harmonicity. Otherwise, the *l*-th frame is modified so that the length of the frame is only one single pitch period. In other words, the modified signal of the *l*-th frame, $\hat{s}^{(l)}(n)$, is represented as

$$\hat{s}^{(l)}(n) = s^{(l)}(n)\ w\left(\frac{N}{2} - \frac{P^{(l)}}{2} - n\right) \tag{2}$$

where $P^{(l)}$ is the pitch period of the *l*-th frame and is calculated as $P^{(l)} = \lfloor f_s / F_0^{(l)} + 1/2 \rfloor$ with a ceiling operator of $\lfloor \cdot \rfloor$. In addition, $w(n)$ is a Hanning window with a length $P^{(l)}$, *i.e.*, $w(n) = 0.5 - 0.5\cos(2\pi n / P^{(l)} - 1)$. In other words, the harmonic component is obtained by taking one pitch period including the center of a frame. On the other hand, when $F_0^{(l)} = 0$, *i.e.*, when the frame includes mostly an unvoiced signal, $s^{(l)}(n)$ is modified as

$$\hat{s}^{(l)}(n) = s^{(l)}(n)\ w\left(\frac{N}{4} - n\right) \tag{3}$$

where the length of $w(n)$ is $N/2$ such that $w(n) = 0.5 - 0.5\cos(4\pi n / N - 1)$. Next, as in (2) or (3), the modified signal is used for estimating the harmonic and non-harmonic components.

An estimate of the harmonic component, $\bar{s}_h^{(l)}(n)$, can be obtained by applying a harmonic spectral shaping filter, $h_h^{(l)}(n)$, as

$$\bar{s}_h^{(l)}(n) = \hat{s}^{(l)} * h_h^{(l)}(n) \tag{4}$$

where $h_h^{(l)}(n)$ is modeled as

$$h_h^{(l)}(n) = \begin{cases} G_h^{(l)} h_{low}^{(l)}(n), & if \quad F_0^{(l)} > 0 \\ 0, & if \quad F_0^{(l)} = 0 \end{cases} \tag{5}$$

In the above equation, $G_h^{(l)}$ is a gain parameter, and $h_{low}^{(l)}(n)$ is a low-pass filter whose cut-off frequency is the maximum voiced frequency (MVF) of the *l*-th frame, $F_{mvf}^{(l)}$ [19]. The MVF is defined as the frequency at which the change in the power spectrum is maximized. Thus, $F_{mvf}^{(l)}$ is defined as

$$F_{mvf}^{(l)} = \frac{f_s}{2N} \arg\max_{0<k<\frac{N}{2}} \left(\left| \hat{S}^{(l)}(k) \right|^2 - \left| \hat{S}^{(l)}(k-1) \right|^2 \right) \tag{6}$$

where $\hat{S}^{(l)}(k)$ is the Fourier transform of $\hat{s}^{(l)}(n)$. Note that the convolution of (4) is actually performed in the frequency domain as $\bar{S}_h^{(l)}(k) = \hat{S}^{(l)}(k) H_h^{(l)}(k)$, where $H_h^{(l)}(k)$ is the Fourier transform of $h_h^{(l)}(n)$. The gain parameter, $G_h^{(l)}$, is estimated based on power matching between $\hat{s}^{(l)}(n)$ and $\hat{s}^{(l)}(n) * h_{low}^{(l)}(n)$, such that

$$G_h^{(l)} = \frac{\sum_{k=0}^{N/2} \left| \hat{S}^{(l)}(k) \right|^2}{\sum_{k=0}^{N/2} \left| \hat{S}^{(l)}(k) H_{low}^{(l)}(k) \right|^2} \tag{7}$$

where $H_{low}^{(l)}(k)$ is the Fourier transform of $h_{low}^{(l)}(n)$.

Similarly, an estimate of the non-harmonic component, $\bar{s}_{nh}^{(l)}(n)$, can be obtained by applying a non-harmonic spectral shaping filter, $h_{nh}^{(l)}(n)$, as

$$\bar{s}_{nh}^{(l)}(n) = \hat{s}^{(l)} * h_{nh}^{(l)}(n) \tag{8}$$

where $h_{nh}^{(l)}(n)$ is also modeled as

$$h_{nh}^{(l)}(n) = \begin{cases} G_{nh}^{(l)} h_{high}^{(l)}(n) * r(n), & if \ F_0^{(l)} > 0 \\ G_{nh}^{(l)} r(n), & if \ F_0^{(l)} = 0 \end{cases} \tag{9}$$

where $h_{high}^{(l)}(n) = (-1)^n h_{low}^{(l)}(n)$ and $r(n)$ is a random variable that follows a zero-mean and unit-variance Gaussian distribution. In addition, $G_{nh}^{(l)}$ is a gain parameter estimated as

$$G_{nh}^{(l)} = \begin{cases} \dfrac{\sum_{k=0}^{N/2} \left| \hat{S}^{(l)}(k) \right|^2}{\sum_{k=0}^{N/2} \left| \hat{S}^{(l)}(k) H_{high}^{(l)}(k) R(k) \right|^2}, & if \ F_0^{(l)} > 0 \\ \dfrac{\sum_{k=0}^{N/2} \left| \hat{S}^{(l)}(k) \right|^2}{\sum_{k=0}^{N/2} \left| \hat{S}^{(l)}(k) R(k) \right|^2}, & if \ F_0^{(l)} = 0 \end{cases} \tag{10}$$

where $R(k)$ and $H_{high}^{(l)}(k)$ are the Fourier transforms of $r(n)$ and $h_{high}^{(l)}(n)$, respectively. Finally, $\bar{s}^{(l)}(n)$, which is an estimate of $s^{(l)}(n)$, is obtained by adding $\bar{s}_h^{(l)}(n)$ and $\bar{s}_{nh}^{(l)}(n)$ using a synchronous and asynchronous overlap-and-add method [17].

As described above, the HNH model is composed of five feature parameters: $F_0^{(l)}$, $F_{mvf}^{(l)}$, $G_h^{(l)}$, $G_{nh}^{(l)}$, and $\hat{S}^{(l)}(k)$. In fact, instead of directly using $\hat{S}^{(l)}(k)$ for the HNH model, i.e., 257 frequency bins, 24 mel-frequency cepstral coefficients (MFCCs) are extracted from $\hat{S}^{(l)}(k)$ [14]. By doing this, the complexity of the speaker-adaptive training and speech synthesis can be reduced, and the performance of the speaker-adaptive training can be improved [20]. Moreover, it is known that a loss of the excitation components in $\hat{S}^{(l)}(k)$ through an MFCC analysis does not greatly influence the quality of synthesized speech since the major excitation components are recovered by the speech synthesis [17].

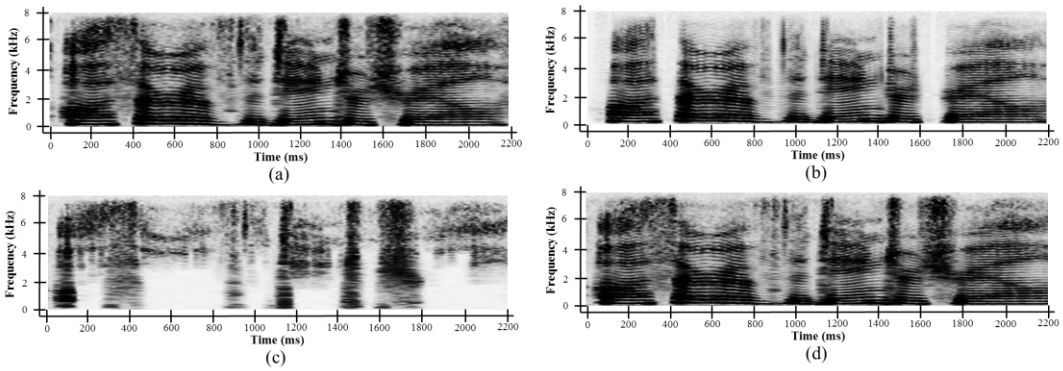


Figure 2. Illustration of the HNH Modeling: Spectrograms of (a) Reference Speech Signal, (b) Synthesized Harmonic Component by the HNH Model, (c) Synthesized Non-harmonic Component by the HNH Model, and (d) Synthesized Speech Signal by the HNH Model

Figure 2 illustrates the performance of HNH modeling. For a given reference speech signal (Figure 2(a)), $\bar{s}_h^{(l)}(n)$, $\bar{s}_{nh}^{(l)}(n)$, and $\bar{s}^{(l)}(n)$ are shown in Figures 2(b), 2(c), and 2(d), respectively. It can be seen from the figure that the spectrogram of the synthesized speech from the HNH model is almost identical to that of the reference speech.

Table 1. Comparison of the Feature Type and Number of Feature Parameters per Frame for the STRAIGHT and HNH Models

Method	Feature Type	No. of Feature Parameters	Total
STRAIGHT	Log F0	1	30
	Spectrum (MFCC)	24	
	Band Aperiodicity	5	
HNH Model	Log F0	1	28
	Spectrum (MFCC)	24	
	Maximum voiced frequency	1	
	Gain	2	

3.1.2. Parameter Quantization and Transmission for the Client-Server Approach: Table 1 lists the number of feature parameters, and compares this number with that of the speech transformation and representation using an adaptive interpolation of the weighted spectrum (STRAIGHT) [21]. As shown in the table, the HNH model represents a non-harmonic speech component using one MVF parameter and one gain parameter, while STRAIGHT uses five parameters for modeling the aperiodicity of the speech. Consequently, the HNH model reduces the total number of feature parameters by 6.7%, compared to STRAIGHT. Such a reduction not only improves the performance of the speaker-adaptive training [22] but also reduces the amount of transmission data from the client to the server required by this client-server framework.

To transmit the 28-dimensional feature parameters, the feature parameters are quantized, where a multi-stage predictive split vector quantization scheme proposed in [23] is used. In other words, the quantizer first utilizes a linear prediction to take advantage of the correlations among the subsequent feature vectors. Next, the error vector from the linear prediction is subjected to the first stage of the multi-stage split vector quantization. In other words, the first stage selects an entry in a codebook that best approximates the error vector by

measuring the Euclidean distance. Here, 18 bits are assigned to the first stage. In the second stage, the 28-dimensional residual vector is split into seven sub-vectors from $\mathbf{c}_1$ to $\mathbf{c}_7$ with dimensions of 8, 8, 8, 1, 1, 1, and 1, where the first three sub-vectors correspond to the MFCCs, and the remaining four scalars correspond to the pitch, MVF, and two gain parameters, respectively. Similar to the first stage of the vector quantization, each sub-vector is quantized by selecting an entry from the corresponding codebook that best approximates the error vector by measuring the Euclidean distance. In this stage, 16 bits are assigned for the 8-dimensional sub-vector quantization, while 6 bits each are used for quantizing the pitch, MVF, and gain parameters. As a result, the 28-dimensional feature parameters are quantized using 90 bits in total, as shown in Table 2.

Table 2. Bit Allocation for a Multi-Stage Predictive Split Vector Quantization of the HNH Model Parameters

Stage	Feature	Sub-Vector	Dimension	Allocated Bits	Total Bits
1	-	$\{\mathbf{c}_1 - \mathbf{c}_7\}$	28	18	90
2	Spectrum (MFCC)	$\mathbf{c}_1$	8	16	
		$\mathbf{c}_2$	8	16	
		$\mathbf{c}_3$	8	16	
	Log F0	$\mathbf{c}_4$	1	6	
	MVF	$\mathbf{c}_5$	1	6	
	Harmonic gain	$\mathbf{c}_6$	1	6	
	Non-harmonic gain	$\mathbf{c}_7$	1	6	

Since the frame shift is 2.5 ms, the number of feature vectors becomes 400 per second, resulting in a fixed data rate of 36 kbit/s.

After the feature vectors are quantized, they are transmitted from the client to the server. In the server, the transmitted vector is then inversely quantized and converted back into 28-dimensional feature parameters. These parameters are applied to speaker-adaptive training.

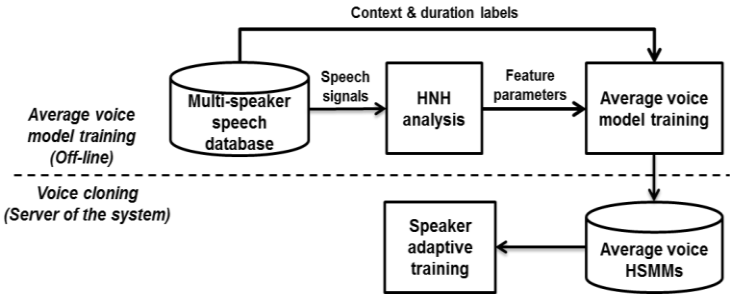


Figure 3. Block Diagram of the Average Voice Model Training

3.1.3. Average Voice Model and Speaker-Adaptive Training: Before performing the speaker-adaptive training to obtain the acoustic and duration models of the target speaker, the average voice models must be prepared. Figure 3 shows a block diagram of the average voice model training. As shown in the figure, speech signals from various speakers and their labels are required to train the average voice models. In this paper, six hours of speech signals, uttered by six speakers, and their corresponding context and duration labels are taken from the CMU US ARCTIC speech database. Of course, the feature parameters of these speech

signals are extracted using the HNH model. Each average voice model is composed of a context-dependent multi-stream left-to-right multi-space probability distribution (MSD) hidden semi-Markov model (HSMM) [25] to simultaneously model its acoustic features and duration.

Next, speaker-adaptive training is conducted using the target speeches and their labels. Similar to the average voice models, the labels of the target speeches also consist of both context and duration information. It should be noted that context information is invariant to recorded speech because the sentence is known. However, the duration information varies for each recording attempt, and thus duration labeling should be performed for each recorded speech. In the proposed system, the duration labeling is performed using Viterbi decoding using the average voice models [26-27]. Specifically, for a given phoneme, and its state duration densities from the average voice model, λ, the state durations of for the target speaker, are estimated by maximizing the likelihood as $\hat{d} = \arg\max_{d} \{ p(d \mid w, \lambda) \}$.

Table 3. Mean Absolute Error (MAE) and mean Absolute Percentage Error (MAPE) of the Duration Labeling for Six Speakers from the CMU US ARCTIC Speech Database

Speaker (gender)	MAE (ms)	MAPE (%)
AWB (male)	9.14	8.94
BDL (male)	8.95	10.88
CLB (female)	10.17	8.90
JMK (male)	9.17	8.82
RMS (male)	8.99	9.37
SLT (female)	10.87	10.30
Average	9.55	9.54

Finally, the feature parameters and labels of the speeches by the target speaker are used for the speaker-adaptive training. Among the various adaptation techniques, the CSMAPLR adaptation technique [13] is used in this study. This is because the CSMAPLR adaptation technique can provide synthetic speeches close to the target speaker, as com-pared to the CMLLR and SMAPLR adaptation techniques [8].

3.2. Speech Synthesis Stage

In the speech synthesis stage, an input text coming from the client side is sent directly to the server. The text is then analyzed and converted into context labels. The procedure of the text analysis is identical to that used for the speaker-adaptive training. The acoustic parameters corresponding to these labels are extracted from the acoustic models of the target speaker using the parameter generation technique reported in [15]. These acoustic parameters are quantized using the same vector quantizer described in Section 3.1.1, and then transmitted back to the client. Finally, the speech signals are synthesized using the transmitted parameters.

4. Performance Evaluation

The performance of the proposed HMM-based distributed TTS system was evaluated for the voice cloning and speech synthesis stages separately. In particular, the performance of the voice cloning stage was measured in terms of the duration labeling accuracy and processing time. Meanwhile, as the performance of the speech synthesis stage, the subjective quality of

the synthesized speeches was measured so as to compare the similarity between the uttered speeches by the target speaker and the synthesized speeches adapted to the target speaker. The intelligibility of the synthesized speeches was also measured.

The experimental environments were as follows. First, the client side was implemented using a laptop with a single-core processor at a clock speed of 1.6 GHz. Moreover, the speeches of the target speaker were recorded using an electret condenser microphone embedded in the laptop. Second, the server was implemented using a high-end workstation with a quad-core processor at a clock speed of 3.2 GHz. In addition, the average voice HMMs in the voice cloning stage were prepared using the HTK toolkit for speech synthesis (HTS) version 2.1 [28] with six-hour speeches from four males and two females each, which were selected from the CMU US ARCTIC speech database. Finally, the client and server were connected through a wireless network.

Table 4. Comparison of the Processing Time (ms) between a Client-Based TTS System and the Proposed Distributed TTS System

Processing step	Client-Based TTS System	Distributed TTS System	
		Client	Server
HNH analysis	1.16	1.16	-
Feature compression	0.00	1.02	-
Feature decompression	0.00	-	0.44
Duration labeling	4.65	-	0.87
Speaker adaptive training	19.84	-	6.20
Total	25.65	7.51	

4.1. Evaluation of Voice Cloning Stage

First, the accuracy of the duration labeling was evaluated using the mean absolute error (MAE), which was defined as the average difference in duration obtained between manual labeling and automatic labeling as described in Section 3.1.2. To this end, 100 speeches and their corresponding labels from four males and two females each were selected from the CMU US ARCTIC speech database. The differences in duration were obtained by calculating the absolute values of the difference between the manually labeled duration and the automatically labeled duration for each phoneme. In addition, the mean absolute percentage error (MAPE) was also computed. Table 3 shows the MAE and MAPE of the duration labeling, where all of the numbers were averaged over all phonemes in the test data. It was shown from the table that the MAE and MAPE were measured as 9.55 ms and 9.54%, respectively, which was acceptable for application in speaker-adaptive training.

Next, the processing time of the proposed distributed TTS system was compared with that of a typical TTS system, which was operated only on the client side. As previously mentioned, the processing steps of the distributed TTS system were distributed to balance the computational complexity between the client and server. As shown in Table 4, the HNH analysis and feature compression of the proposed distributed TTS system were performed on the client, whereas feature decompression, duration labeling, and speaker adaptive training were performed on the server. Thus, the processing time required for the proposed distributed TTS system could be measured as the shorter one between the client and server. Consequently, it was shown from the table that the processing time of the proposed distributed TTS system was 3.4-times smaller than that of the client-based TTS system.

4.2. Evaluation of Speech Synthesis Stage

The performance of the speech synthesis stage was measured through subjective tests. First, the effect of the proposed distributed TTS system on the speech quality was investigated since the HNH model parameters should be quantized for client-server processing. Thus, two sets of synthesized speeches were obtained using the model parameters with and without the quantization. The quality similarity between the synthesized speeches with and without the quantization was then measured using the perceptual evaluation of speech quality (PESQ) [24]. It was shown from the measurement that an average PESQ score was 4.39 for twenty sentences, and thus the distortion from the quantization of the HNH model parameters was negligible.

Next, the performance of the proposed TTS system was compared with that of a conventional TTS system with speaker-dependent training [29]. To this end, we prepared twelve sets of HSMMs for six speakers: six sets were for the conventional TTS system and the other six sets were for the proposed TTS system. Note that for each speaker the conventional TTS system used one hour of speech signals of the speaker for the speaker-dependent training, whereas the proposed TTS system used 10 min of speech signals for the speaker-adaptive training. After the training was completed, the twenty sentences were applied to both TTS systems to synthesize the speeches of the target speakers.

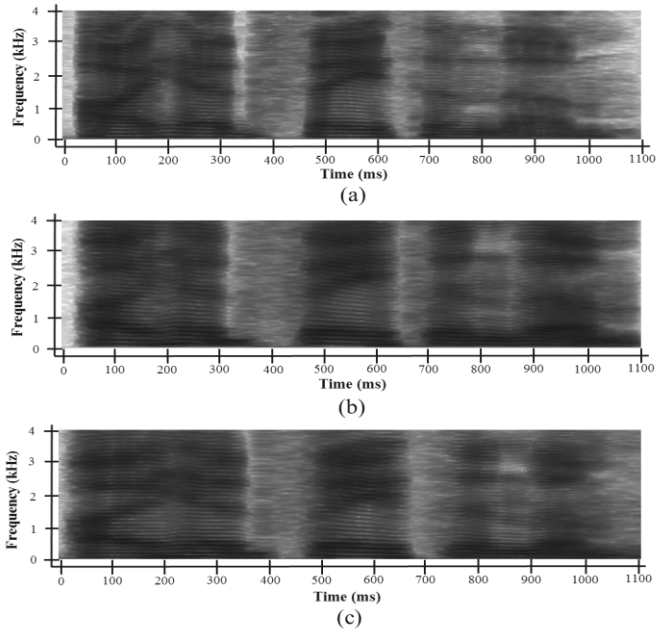


Figure 4. Comparison of Spectrograms of (a) an Uttered Speech by Speaker C, (b) a Synthesized Speech from a Conventional TTS system, and (c) a Synthesized Speech from the pro-posed TTS System

Figure 4 shows a spectrogram of an uttered speech by one of the speakers, as well as spectrograms of the speeches synthesized by the conventional and proposed TTS systems. It was shown from the figure that the proposed TTS system provided a synthesized speech similar to a conventional TTS system even when the acoustic models of the proposed TTS system were trained using a much smaller amount of speech data than the conventional TTS system.

Finally, listening tests were conducted to determine the intelligibility and similarity between the uttered and synthesized speech signals by both the conventional and proposed TTS systems. Eight participants listened to the speech samples and were asked to give a mean opinion score (MOS) ranging from 0 to 5 regarding the intelligibility of the synthesized speech and the similarity between the uttered speech samples and their synthesized versions.

Table 5. Comparison of the Mean Opinion Scores for the Intelligibility and Similarity of the Synthesized Speeches by the Conventional and Proposed Speaker-Dependent TTS Systems

Speaker	Conventional TTS System		Proposed TTS System	
	Intelligibility	Similarity	Intelligibility	Similarity
A	3.70	3.57	3.60	3.63
B	3.52	3.74	3.54	3.83
C	3.49	4.10	3.40	4.07
D	3.45	4.05	3.47	4.09
E	3.83	3.88	3.75	3.95
F	3.68	3.33	3.66	3.41
Average	3.61	3.78	3.47	3.76

Table 5 compares the average MOS for the intelligibility and similarity of the synthesized speeches from the conventional and proposed TTS systems for the six speakers. As shown in the table, the proposed TTS system achieved an average MOS comparable to the conventional TTS system for both intelligibility and similarity. This implies that a new speaker's voice could be synthesized using a smaller amount of adaptation data than that required by the conventional speaker-dependent TTS system.

5. Conclusion

In this paper, an HMM-based distributed TTS system was proposed in a client-server framework to reduce the computational complexity of speech synthesis using speaker-adaptive training. In particular, the proposed system distributed processing modules to both the client and the server. Thus, lower complexity processing modules, such as feature extraction using HNH modeling, feature compression, and speech synthesis, were processed on the client, while feature decompression, text analysis, and speaker-adaptive training were performed on the server. To demonstrate the effectiveness of the proposed system, the processing time of the proposed distributed TTS system was first compared with that of a client-based TTS system. It was shown from the comparison results that the distributed TTS system worked 3.4 times faster than the client-based TTS system. The performance of the proposed TTS system was next compared with that of a conventional TTS system with speaker-dependent training. A comparison of the spectrograms and informal listening tests revealed that a new speaker's voice could be synthesized by the proposed system using a smaller amount of adaptation data than required by a conventional speaker-dependent TTS system. In addition, the proposed TTS system provided a synthesized speech quality similar to a conventional TTS system.

Acknowledgements

This study was (partially) supported by Seoul National University of Science and Technology.

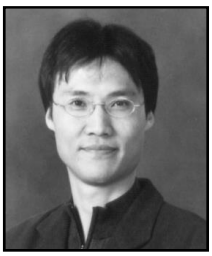
References

[1] S. M. Kim and H. K. Kim, "Hybrid Probabilistic Adaptation Mode Controller for Generalized Sidelobe Cancellers Applied to Multi-microphone Speech Enhancement, Digital Signal Processing, vol. 25, (**2014**) February, pp. 123-133.
[2] S. M. Kim and H. K. Kim, "Probabilistic Spectral Gain Modification Applied to Beamformer Based Noise Reduction in a Car Environment", IEEE Transactions on Consumer Electronics, vol. 57, no. 2, (**2011**) May, pp. 866-872.
[3] K. M. Jeon, N. I. Park, H. K. Kim, M. K. Choi and K. I. Hwang, "Mechanical Noise Suppression based on Non-negative Matrix Factorization and Multi-band Spectral Subtraction for Digital Cameras", IEEE Transactions on Consumer Electronics, vol. 59, no. 2, (**2013**) May, pp. 296-302.
[4] S. M. Kim, H. K. Kim, S. J. Lee and Y. Lee, "Multiple Likelihood Ratio Test-Based Voice Activity Detection Robust to Impact Noise in a Car Environment", Information: an International Interdisciplinary Journal, vol. 16, no. 3, (**2013**) March, pp. 3141-3151.
[5] N. I. Park and H. K. Kim, "Artificial Bandwidth Extension of Narrowband Speech Applied to CELP-type Speech Coding", Information: an International Interdisciplinary Journal, vol. 16, no. 3B, (**2013**) March, pp. 3153-3164.
[6] J. A. Kang and H. K. Kim, "Adaptive Redundant Speech Transmission over Wireless Multimedia Sensor Networks based on Estimation of Perceived Speech Quality", Sensors, vol. 11, no. 9, (**2011**) September, pp. 8469-8484.
[7] Y. R. Oh and H. K. Kim, "A Hybrid Acoustic and Pronunciation Model Adaptation Approach for Non-native Speech Recognition", IEICE Transactions on Information and Systems, E93-D(9), (**2010**) September, pp. 2379-2387.
[8] J. Yamagishi, T. Kobayashi, M. Tachibana, K. Ogata, and Y. Nakano, "Model Adaptation Approach to Speech Synthesis with Diverse Voices and Styles", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Honolulu, Hawaii, (**2007**) April, pp. 1233-1236.
[9] E. Eide, A. Aaron, R. Bakis, W. Hamza, M. Picheny and J. Pitrelli, "A Corpus-based Approach to Expressive Speech Synthesis", Proceedings of ISCA Speech Synthesis Workshop (SSW), Pittsburgh, PA, (**2004**) June, pp. 79-84.
[10] J. Yamagishi, T. Nose, H. Zen, Z. Ling, T. Toda, K. Tokuda, S. King and S. Renals, "Robust Speaker-adaptive HMM-based Text-to-speech Synthesis", IEEE Transactions on Audio, Speech and Language Processing, vol. 17, no. 6, (**2009**) August, pp. 1208-1230.
[11] M. J. F. Gales, "Maximum Likelihood Linear Transformations for HMM-based Speech Recognition", Computer Speech and Language, vol. 12, no. 2, (**1998**) April, pp. 75-98,
[12] O. Shiohan, T. A. Myrvoll and C. H. Lee, "Structural Maximum A Posteriori Linear Regression for Fast HMM Adaptation", Computer Speech and Language, vol. 16, no. 1, (**2002**) January, pp. 5-24.
[13] Y. Nakano, M. Tachibana, J. Yamagishi and T. Kobayashi, "Constrained Structural Maximum a Posteriori Linear Regression for Average-voice-based Speech Synthesis", Proceedings of International Conference on Spoken Language Processing (ICSLP), Pittsburgh, PA, (**2006**) September, pp. 2286-2289.
[14] K. Tokuda, H. Zen, and A. W. Black, "An HMM-Based Speech Synthesis System Applied to English", Proceedings of IEEE Speech Synthesis Workshop (SSW), Santa Monica, CA, (**2002**) September, pp. 227-230.
[15] K. Tokuda, T. Kobayashi and S. Imai, "Speech Parameter Generation from HMM Using Dynamic Features. Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Philadelphia, PA, (**2005**) March, 660-663.
[16] J. Kominek, and A. W. Black, "The CMU ARCTIC Speech Databases. Proceedings of ISCA Speech Synthesis Workshop (SSW), Pittsburgh, PA, (**2004**) June, pp. 223-224.
[17] K. M. Jeon, "Harmonic and Non-harmonic Modeling of Speech for Statistical Parametric Speech Synthesis. MS Thesis, Gwangju Institute of Science and Technology, Gwangju, Korea, (**2012**) January.
[18] D. Talkin, "Speech Coding and Synthesis", Elsevier Science B. V., Amsterdam, Netherlands, (**1995)**
[19] Y. Stylianou, J. Laroche and E. Moulines, "High-quality Speech Modification based on a Harmonic + Noise Model. Proceedings of European Conference on Speech Communication and Technology (Eurospeech), Madrid, Spain, (**1995**) September, pp. 451-454.
[20] A. K. Jain, "Statistical Pattern Recognition: A Review", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, no. 1, (**2000**) January, pp. 3-37.
[21] H. Kawahara, I. Masuda-Katsuse and A. Cheveigné, "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, nos. 3-4, (**1999**) April, pp. 187-207.

[22] J. Yamagishi, T. Nose, H. Zen, T. Toda and K. Tokuda, "Performance Evaluation of the Speaker-independent HMM-based Speech Synthesis System "HTS 2007" for the Blizzard Challenge 2007", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Las Vegas, NV, (**2008**) March, pp. 3957-3960.
[23] G. N. Ramaswamy and P. S. Gopalakrishnan, "Compression of Acoustic Features for Speech Recognition in Network Environments", Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Seattle, WA, May (**1998**), pp. 977-980.
[24] ITU-T Recommendation P.862, Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-band Telephone Networks and Speech Codecs, (**2001**) January.
[25] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi and T. Kitamura, "A Hidden Semi-Markov Model-based Speech Synthesis System", IEICE Transactions on Information and Systems, E90-D(5), (**2007**) May, pp. 825-834.
[26] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", Proceedings of the IEEE, vol. 77, no. 2, (**1989**) February, pp. 257-286.
[27] S. J. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland, "The HTK book (for HTK Version 3.4). Cambridge University Press", Cambridge, U.K., (**2006)** December**.**
[28] H. Zen, K. Oura, T. Nose, J. Yamagishi, S. Sako, T. Toda, T. Masuko, A. W. Black and K. Tokuda, "Recent Development of the HMM-based Speech Synthesis System", Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference, Sapporo, Japan, (**2009**) October, pp. 121-130.
[29] H. Zen and T. Toda, "An Overview of Nitech HMM-Based Speech Synthesis System for Blizzard Challenge 2005", Proceedings of European Conference on Speech Communication and Technology (Eurospeech), Lisbon, Portugal, (**2005**) September, pp. 93-96.

Authors

Kwang Myung Jeon, he received a B.S. degree in Information and Communications Engineering from Sejong University in 2010 and an M.S. degree in Information and Communications Engineering from the Gwangju Institute of Science and Technology (GIST), Korea, in 2012. He is currently pursuing a Ph.D. at GIST. His current research interests include speech and audio denoising, speech synthesis, and embedded algorithms and solutions for speech and audio processing for handheld devices.



Seung Ho Choi, he received a B.S. degree in Electronic Engineering from Hanyang University, Korea in 1991. He then received both M.S. and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology (KAIST), Korea in 1993 and 1999, respectively. He was a senior researcher at Samsung Advanced Institute of Technology, Korea, from 1996 to 2002. He was a visiting professor at University of Florida, USA, from 2008 to 2009. Since August 2002, he has been with the Department of Electronic and IT Media Engineering at Seoul National University of Science and Technology as a professor. His current research interests include speech and audio coding, acoustic signal processing, speech recognition.