# Comparison of Tied-Mixture and State-Clustered HMMs with Respect to Recognition Performance and Training Method

*Hiroyuki Segi, NHK (Japan Broadcasting Corporation) Science & Technology Research Laboratories, Fukushima, Japan*

*Kazuo Onoe, NHK (Japan Broadcasting Corporation), Science & Technology Research Laboratories, Fukushima, Japan*

*Shoei Sato, NHK (Japan Broadcasting Corporation) Science & Technology Research Laboratories, Fukushima, Japan*

*Akio Kobayashi, NHK (Japan Broadcasting Corporation) Science & Technology Research Laboratories, Fukushima, Japan*

*Akio Ando, Faculty of Engineering, University of Toyoma, Toyoma, Japan*

## ABSTRACT

*Tied-mixture HMMs have been proposed as the acoustic model for large-vocabulary continuous speech recognition and have yielded promising results. They share base-distribution and provide more flexibility in choosing the degree of tying than state-clustered HMMs. However, it is unclear which acoustic models to superior to the other under the same training data. Moreover, LBG algorithm and EM algorithm, which are the usual training methods for HMMs, have not been compared. Therefore in this paper, the recognition performance of the respective HMMs and the respective training methods are compared under the same condition. It was found that the number of parameters and the word error rate for both HMMs are equivalent when the number of codebooks is sufficiently large. It was also found that training method using the LBG algorithm achieves a 90% reduction in training time compared to training method using the EM algorithm, without degradation of recognition accuracy.*

*Keywords:* *Acoustic Model, EM Algorithm, LBG Algorithm, Speech Database, Speech Recognition, State-Clustered HMM, Tied-Mixture HMM, Training Method*

DOI: 10.4018/jitr.2014070102

## 1. INTRODUCTION

Speech-recognition systems are of particular interest in Japan because real-time keyboard entry in the Japanese language is complicated by the need to select the correct characters among homonyms.

Remarkable advances have been made in speech-recognition technology in recent years. One example is the simultaneous subtitling system for Japanese television broadcast programs developed by Kobayashi (2013), which uses speech recognition to make real-time captions for use by the hearing impaired. Another example is the transcription system using speech recognition developed by Kawahara (2012) that is currently deployed in the Japanese Parliament.

These systems employ Hidden Markov Models (HMMs) that were proposed long time ago and many speech recognition systems are still using HMMs even now (Hofmann, 2012; Liu, 2013; Ogawa, 2012; Singh, 2012; Siu, 2012). The continuing development of large-scale speech databases has made it possible to use large amounts of data to train HMMs (Itou, 1998; Maekawa, 2000; Segi, 2010). As the volume of data increases, it is possible to increase the number of parameters without losing the estimation accuracy, and highly accurate speech recognition can be realized by introducing more complex structures in HMMs. For example, a state-clustered HMM (Hwang, 1996; Onoe, 2003; Young, 1994) has been proposed in which base (usually Gaussian) distributions and weights are shared within individual clusters. In addition, a tied-mixture HMM (Nguyen, 1995; Sankar, 1998; Lee, 2000), in which base distributions and weights can be shared separately, has been reported to produce favorable results.

However, the performance of different acoustic models has not yet been compared using the same training data. Moreover, the established training methods for HMMs, the Linde-Buzo-Gray (LBG) algorithm (Linde, 1980) and the Expectations-Maximization (EM) algorithm (Dempster, 1977), have not been compared. Although these training methods are proposed long time ago and discriminative training (Povey, 2002) is used in recent years, these training methods are used now to make initial models for discriminative training (Delcroix, 2013).

The current paper compares the recognition performance of the respective HMMs and training methods under the same conditions. Section 2 describes previous comparisons of state-clustered and tied-mixture HMMs. Section 3 describes the differences between state-clustered and tied-mixture HMMs. Section 4 describes HMM training methods. Section 5 discusses recognition tests as follows: section 5.1 describes test conditions; section 5.2 compares training methods for state-clustered and tied-mixture HMMs; section 5.3 compares recognition performance and processing speeds with different numbers of tied-mixture HMM weights; section 5.4 compares state-clustered and tied-mixture HMMs with different numbers of base distributions included in the codebook; and section 5.5 compares state-clustered and tied-mixture HMMs with the same overall number of base distributions but different codebook sizes. Section 6 summarizes this work.

## 2. PREVIOUS WORK

State-clustered and tied-mixture HMMs have been compared in previous studies. Huang (1993) showed that a tied-mixture HMM had better recognition performance than a state-clustered HMM, although the training method differed between the two. Sankar (1998) compared a state-clustered HMM (with 937 clustered states and 32 base distributions per state) with a tied-mixture HMM (with 38 codebooks and 789 base distributions per codebook), and found that when the overall numbers of base distributions was the same, the latter had better recognition performance. Digalakis (1994) found that a tied-mixture HMM with 495 code-

books showed better recognition performance that those with 1 or 40 codebooks, even though the number of total parameters was smaller.

It has not been established whether this difference in recognition performance originates from differences in the tied structures of the base distributions between states, from the increased number of base distributions per codebook, or from the increased number of combinations of different weights caused by the switch to a tied-mixture HMM. To clarify whether the difference is attributable to an increase in the number of parameters or an increase in structural complexity, we compared recognition performance using the same base distribution shared structures and the same number of base distributions per codebook between state-clustered and tied-mixture HMMs.

Moreover, state-clustered and tied-mixture HMMs are created in different ways, and the training methods have not previously been compared. The standard procedure for creating a state-clustered HMM is to repeat the EM algorithm a number of times whenever a change is made to the HMM structure, such as increasing the number of mixtures. By contrast, the standard procedure for creating a tied-mixture HMM involves partitioning the pre-aligned training data into multiple clusters using the LBG or k-means algorithm, and then using them for direct calculation of the HMM, after which the EM algorithm is repeated a number of times.

Here we compared recognition performance using the same training methods to create state-clustered and tied-mixture HMMs, as well as comparing the EM and LBG algorithms.

## 3. STATE-CLUSTERED HMM AND TIED-MIXTURE HMM

To see the differences between HMM structures, it is helpful to understand how the structure of each HMM is determined. Methods for determining the structure of HMMs are discussed below.

### 3.1. Determining the Structure of a State-Clustered HMM

1. We create a no-tied triphone HMM. We studied two such methods: (i) creation by repeated application of the EM algorithm from a monophone initial model, and (ii) creation from training data with alignment;
2. First no-tied HMMs that have the same central phoneme and state position are collected together. Next they are bisected into those that correspond to questions and those that do not, and the question for which the largest difference $\Delta L(q)$ in log-likelihood $L(C)$ after bisection is used as the question for the initial decision tree:

$$L(C) = -\frac{1}{2}\left(\log\left((2\pi)^n\left|\Sigma(C)\right|\right) + n\right)\left(\sum_{s\in C}\sum_{f\in F}\gamma_s(O^f)\right) \quad (1)$$

$$\Delta L(q) = L(C(q,YES)) + L(C(q,NO)) - L(C) \quad (2)$$

Here, $\Sigma(C)$ is the variance matrix of the tied state $C$, $O^f$ is the feature vector observed in the $f$-th frame, $n$ is the dimensionality of the feature vector, $F$ is the set of training frames, and $\gamma_s(O^f)$ is the probability of state occupancy in a state $s$. $C(q, YES)$ is a tied state that is classified as YES for question $q$, and $C(q, NO)$ is a tied state that is classified as NO for question $q$.

Question $q$ normally asks whether or not a phoneme has the same articulation position or phonological representation method, such as "Is the previous phoneme fricative?", or "Is the following phoneme a consonant?". For the test of section 4, we prepared approximately 100 types (Segi, 1999).

$\Sigma(C)$ can be calculated by the following equation:

$$\Sigma(C) = \frac{\sum_{s\in C}\left(\Sigma_s + \mu_s\mu_s^{\mathrm{T}}\right)\left(\sum_{f\in F}\gamma_s(O^f)\right)}{\sum_{s\in C}\left(\sum_{f\in F}\gamma_s(O^f)\right)} - \mu(C)\mu(C)^{\mathrm{T}} \quad (3)$$

where $\mu(C)$ is the average of a tied state $C$, which can be calculated as follows:

$$\mu(C) = \frac{\sum_{s\in C}\mu_s\left(\sum_{f\in F}\gamma_s(O^f)\right)}{\sum_{s\in C}\left(\sum_{f\in F}\gamma_s(O^f)\right)} \quad (4)$$

3. The two HMMs obtained by this bisection are bisected again by different questions, and in the same way as above, the question that causes the largest difference in log-likelihood values before and after bisection is added to the decision tree;
4. This process is repeated until the difference $\Delta L(q)$ in log-likelihood values before and after bisection falls below a preset threshold, thereby yielding a decision tree;
5. This decision tree is applied to all triphone states, and the base distributions and weights are shared between triphone states belonging to the same leaf.

Thus, in a state-clustered HMM, tied states that have the same base distributions will always have the same weights. Figure 1(a) shows the shared structure of a state-clustered HMM.

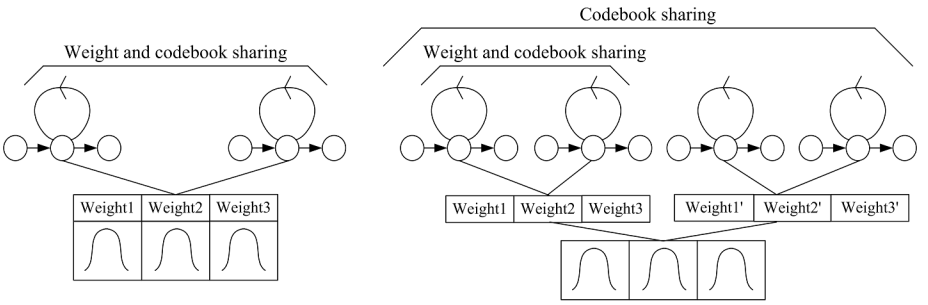
## 3.2. Determining the Structure of a Tied-Mixture HMM

Steps 1 through 5 for a tied-mixture HMM are the same as for a state-clustered HMM. In a tied-mixture HMM, states belonging to the same leaf at step 5 share the same base distribution (codebook), while the way in which weights are shared is determined according to steps 6 and 7:

6. Furthermore, the partitioning by questions is repeated until the difference in log-likelihood values before and after bisection falls below a threshold value that determines the shared structure of weights, thereby yielding a decision tree;
7. This decision tree is applied to all triphone states, and triphone states belonging to the same leaf are treated as a single shared state. That is, not only do states belonging to the same leaf have the same codebook, they also share the weight. This situation is shown in Figure 2.

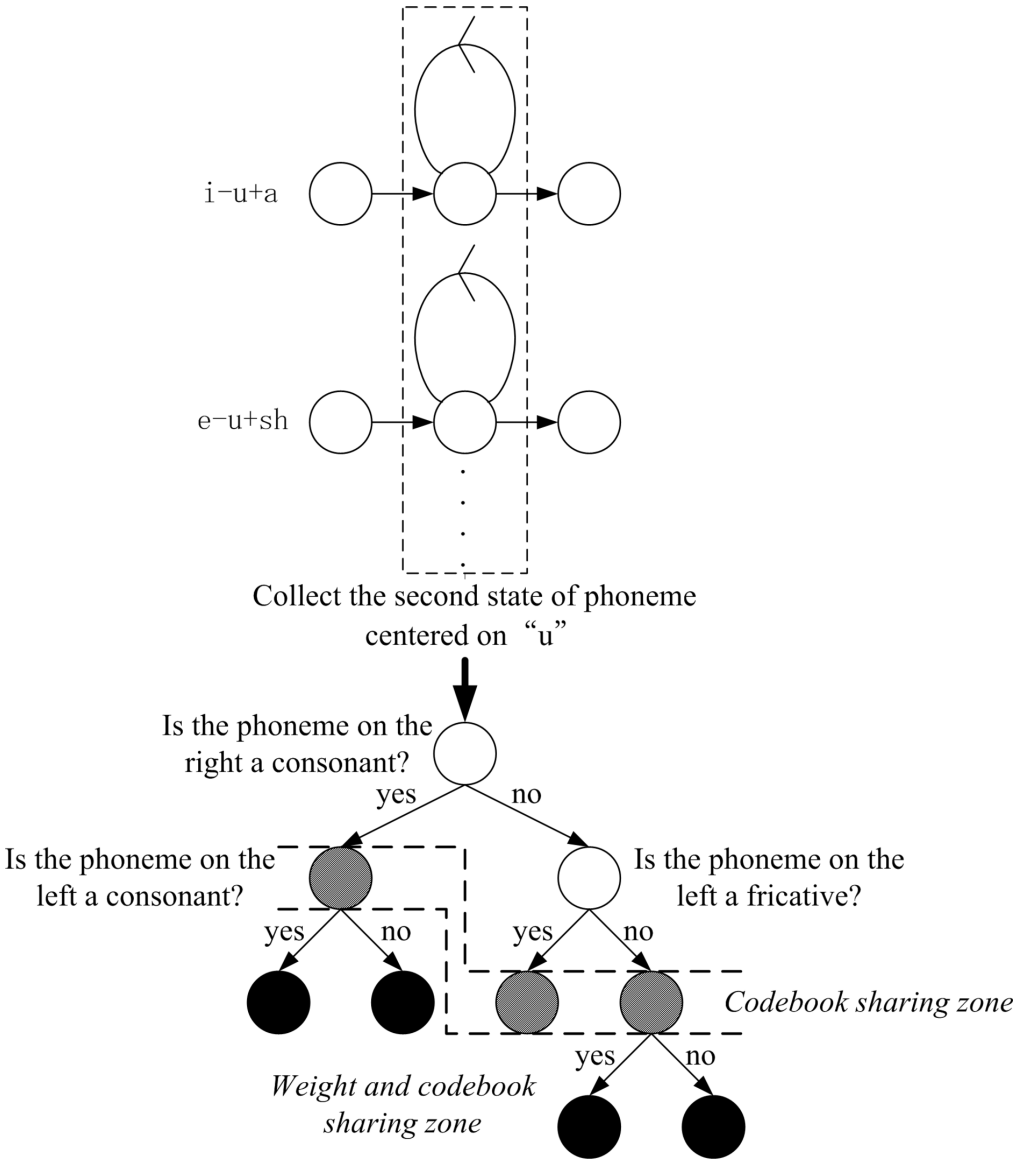
Thus, in a tied-mixture HMM, it is possible to hold a combination of different weights in states that share the same codebook. The shared

*Figure 1. Structure of state-clustered and tied-mixture HMMs*




(a) Structure of state-clustered HMM (b) Structure of tied-mixture HMM

*Figure 2. Weight sharing and codebook sharing*




structure of a tied-mixture HMM is shown in Figure 1(b).

In the recognition test, the sharing relationships of the base distributions were made the same in both models by making the threshold value of the state-clustered HMM equal to the threshold value used for codebook sharing in the tied-mixture HMM. Also, by making the number of base distributions of the same sharing relationships equal to each other, we also made the total number of base distributions the same.

## 4. HMM TRAINING METHODS

When comparing the recognition performance of a state-clustered HMM and a tied-mixture

HMM, there is a possibility that their recognition performance may differ if there are differences in the methods used to create these HMMs. We therefore compared the training method involving repeated application of the EM algorithm, which is normally performed when creating a state-clustered HMM, and the training method involving the LBG algorithm, which is normally performed when creating a tied-mixture HMM.

## 4.1. Training Method 1

From a monophone HMM, create a triphone HMM and perform tree-based clustering. Here, the base distribution and the shared structure of weights are determined. During this process, every time the model is modified, re-estimation by the EM algorithm is performed twice.

Next, the number of base distributions included in the codebook is incremented each time. This increment is achieved by bisecting the base distribution that has the largest sum of weights. The partitioning procedure makes the average value of the base distributions equal to the average value ±(standard deviation)×0.2, with the same variance value and half the weight. After increasing the number of mixtures, re-estimation by the EM algorithm is performed twice in the same way as above. This is repeated until the desired number of mixtures or codebook length is achieved.

## 4.2. Training Method 2

In this method, alignment is performed to obtain the correspondence between each training data item, model and state. In the recognition test described in section 4, we performed alignment with 32 mixed state-clustered HMMs created by method 1.

Next, in each state of each triphone, the mean and variance are calculated from the training data, and tree-based clustering is performed. Here, the base distribution and the shared structure of weights are determined.

Furthermore, the LBG algorithm is applied to training data belonging to each state where the same codebook is shared.

The mean and variance are calculated with the partitioned training data, and these are taken as the mean and variance of the HMM's base distribution. The weights are the values obtained by dividing the number of partitioned training data items by the total number of training data items before they are partitioned. That is, the initial values of the tied-mixture HMM weights are all the same in states where the same codebook is used.

When the training data is partitioned, we require the distance $L(M, O^f)$ between the feature vector $O^f$ observed in the $f$-th frame and the cluster $M$ partitioned by the LBG algorithm, which is obtained by using log-likelihood calculations (See Box 1).

Here, $n$ is the dimensionality of the feature vector, $\mu(M)$ and $\Sigma(M)$ are the mean and variance of the partitioned cluster $M$.

The transition matrix $a_{ij}$ from state $i$ to state $j$ at the phoneme is calculated by the following formula:

$$a_{ij} = \frac{N_{ij}}{N_i} \tag{6}$$

Here, $N_{ij}$ is the number of transition from state $i$ to state $j$ at the phoneme, $N_i$ is the number of transition from state $i$ at the phoneme.

*Box 1.*

$$L(M, O^f) = -\frac{1}{2}\left[\log\left((2\pi)^n \left|\Sigma(M)\right|\right) + \left(O^f - \mu(M)\right)^{\mathrm{T}} \Sigma(M)\left(O^f - \mu(M)\right)\right] \tag{5}$$

Finally, the EM algorithm is used to re-estimate all the parameters included in the HMM five times.

We will now consider the processing times of training methods 1 and 2. In each process, most of the time is devoted to calculating the output probabilities of the base distributions, so we decided to compare the number of calculations needed to calculate the output probabilities.

In training method 1, since the EM algorithm is used to perform concatenated training repeatedly, it is necessary to calculate the output probabilities of however many active states there are. On the other hand, in training method 2, since the LBG algorithm is used to perform pre-alignment and the training data is associated with the model states, it is only necessary to calculate the output probabilities of one specific state. The differences between the EM and LBG algorithms are shown in Figure 3.

Also, when calculating the output probability of a single state in the EM algorithm, assuming there are $J$ base distributions in the codebook, calculating the output probability of the base distribution will require $J$ calculations. On the other hand, in the LBG algorithm, even when the number of clusters $J$ is increased, it is bisected so that the distance calculation can be performed with just two clusters. Accordingly, the distance calculation corresponding to the output probability of the EM algorithm only ever requires two calculations.

For these reasons, the EM algorithm can be used a smaller number of times in training method 2 than in training method 1, allowing the training time to be reduced.

# 5. RECOGNITION TESTS

## 5.1. Test Conditions

### *5.1.1. Acoustic Model*

The acoustic analysis parameters we used are shown in Table 1.

There were a total of 42 phonemes, comprising 10 vowels, 30 consonants, a sentence initial/final silence and a mid-sentence silence. The mid-sentence silence is a single state where there exists a transition from the first state to the last state. The sentence initial/final silence has three states whereby it can not only transition back to itself or to a neighboring state, but can also transition from the first state to the third state or vice versa. The other phonemes are only allowed to transition back to themselves or to neighboring states.

As the training date for the acoustic model, we used NHK's news speech database (Ando, 2003) for the period from June 1996 through May 1998, and balanced sentences (Iso, 1998) spoken by 24 male announcers, making a total of 63,713 sentences with a duration of 152 hours. Both sources used only male speakers, but with utterances not only from announcers

*Figure 3. The difference between EM algorithm and LBG algorithm*

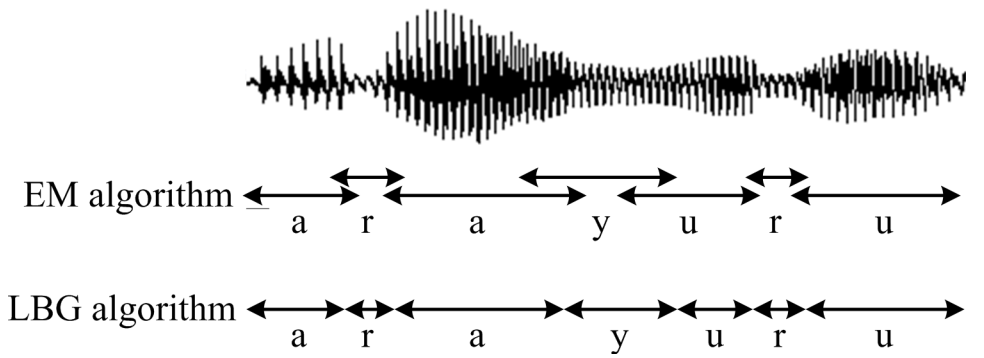


*Table 1. Acoustic analysis parameters*

| | |
|---|---|
| Sampling frequency | 16 kHz |
| Analysis window | 25 ms Hamming |
| Window shift length | 10 ms |
| Analyzed parameters | 12-dimensional MFCC (Mel Frequency Cepstrum Coefficient) + logarithmic power, and its first and second derivatives<br>39 dimensions in total |
| Pre-emphasis | 0.97 |
| Filter banks | 24 |
| Lifters | 22 |

but also from nonprofessional speakers such as reporters. Also, since the training data was recorded and manually transcribed from broadcast news programs, about half of it includes speech data with background sounds such as music or crowd noises.

### 5.1.2. Language Model

As training data for the language model, we used news manuscripts produced by NHK reporters during the period from April 1st 1991 through October 27th 1999 (not including evaluation data). We used the CMU–Cambridge Toolkit to make the language model (Clarkson, 1997). The total number of sentences was 1.7 million, and the total number of words was 68.5 million. For the vocabulary, we used the 60,000 most common words appearing in the news manuscripts. For back-off smoothing, we used the Good–Turing method with a cut-off of 1 for bigrams and 2 for trigrams (Katz, 1987). In this test, we did not use the adapted time-dependent language model (Kobayashi, 1998) of the previous paper.

### 5.1.3. Decoder

We used a two-pass decoder (Imai, 2000). In the first pass, bigrams are used in the language model to perform a Viterbi beam search (Viterbi, 1967) using a tree-structure phoneme network (Ney, 1992). The first-pass search outputs the 200 best sentences, so a history of the four best scores is retained in each state (Schwartz, 1991). In the second pass, the 200 results of the first pass are re-scored using trigrams, and the sentence that achieves the best score is output.

The weights of the acoustic scores and language scores are equalized to 1:14 in both the first and second passes, and the insertion penalty is set to zero.

For the first nodes of words, the maximum bigram values of each word are stored in a table. The language model is also cached to store the bigram values of the 1,500 most recently used words.

### 5.1.4 Evaluation Data

We prepared two types of evaluation data. Test set A consisted of 177 sentences and 6,496 words spoken by a male announcer with no background noise in the morning, midday and evening NHK news broadcasts on 29th September 1999. Test set B consisted of 394 sentences and 7,592 words from the same news programs on the same day that were not included in test set A. Test set B included parts that were spoken by ordinary male speakers and parts containing background noise, and were therefore more acoustically challenging than test set A.

The perplexity (Bahl, 1983), hit rate, and unknown word rate of test set A and B are shown in Table 2.

We decided to evaluate the results of the recognition tests in terms of the word recognition accuracy. The word recognition accuracy is defined by the following formula:

*Table 2. Perplexity, hit rate and unknown word rate of test set*

| | Test Set A | Test Set B |
|---|---|---|
| Bigram perplexity | 84.52 | 202.09 |
| Trigram perplexity | 39.97 | 132.36 |
| Bigram hit rate (%) | 95.56 | 89.97 |
| Trigram hit rate (%) | 75.27 | 57.45 |
| Unknown word rate (%) | 0.74 | 0.91 |

Word recognition accuracy
= (Number of correct words
− Number of incorrectly inserted words)
/ Total number of words

## 5.2. Comparison of HMM Training Methods

As mentioned above, since state-clustered and tied-mixture HMMs are created by different methods, we created HMMs of both types using training methods 1 and 2 as discussed in section 3, and subjected them to recognition tests.

To create the HMMs, the creation times were approximately 15 days with training method 1 and approximately 1.5 days (ten times faster) with training method 2.

The number of codebooks in the HMM created by training method 1 was 1,525 in both cases. The number of weights in the tied-mixture HMM was 5,082. Also, the number of codebooks in the HMM created by training method 2 was 1,543 in both cases. The number of weights in the tied-mixture HMM was 4,876. The number of codebooks and weights differed between training methods because the initial models were constructed differently depending on the training method as discussed in section 3. The overall number of base distributions was roughly 97,000 in both training methods.

The recognition results are shown in Table 3. On comparing the state-clustered HMMs and tied-mixture HMMs, we found no difference in the recognition performance of the state-clustered HMMs or tied-mixture HMMs, either with test set A or with test set B. It can be seen that when a state-clustered HMM is changed into a tied-mixture HMM and the number of weights is increased, it only causes a very small improvement of recognition rates.

On comparing training methods 1 and 2, we decided to create HMMs by training method 2, which has equivalent recognition accuracy but can create HMMs in less time.

Table 3 also shows the word recognition accuracy of a state-clustered HMM created by the method only using the LBG algorithm. By applying the EM algorithm five times, the state-clustered HMM created by training method 2 (LBG algorithm + EM algorithm) is created from the state-clustered HMM created by the method only using the LBG algorithm.

*Table 3. Comparison of HMM training methods*

| | HMM | Test Set A | Test Set B |
|---|---|---|---|
| Training method 1 | State-clustered HMM | 95.89 | 82.01 |
| | Tied-mixture HMM | 96.06 | 82.28 |
| Only using the LBG algorithm | | 95.63 | 81.71 |
| Training method 2 | State-clustered HMM | 95.80 | 81.99 |
| | Tied-mixture HMM | 95.97 | 82.31 |

By comparing the state-clustered HMM by the method only using the LBG algorithm with the state-clustered HMM by training method 2, it can be seen that there are only small difference of the word recognition accuracy between them. Therefore if the tied-mixture HMM is not needed, the training method only using the LBG algorithm can make equal performance compared with the training method 2.

### 5.3. Comparison of Tied-Mixture HMMs with Different Numbers of Weights

By using a constant threshold to determine the shared structure of codebooks in a tied-mixture HMM and varying the threshold used to determine the shared structure of weights, we created five tied-mixture HMMs with different numbers of weights. In each HMM, the number of codebooks was 1,543, and the overall number of base distributions was 97,000. Since there were 1,543 codebooks, the number of tied-mixture HMMs with the smallest number of weights was 1,543, in which case states sharing the same base distributions necessarily have the same weights, so this is identical to a state-clustered HMM.

The recognition results are shown in Figure 4. From Figure 4, it can be seen that the word recognition accuracy for test set A and B are both almost completely invariant with the number of weights. The peak value of word recognition accuracy for the tied-mixture HMM was 95.97% for test set A and 82.31% for test set B. Compared with the values of 95.80% and 81.99% achieved with the state-clustered HMM, this represents an improvement of 0.17% for test set A and 0.32% for test set B. This difference is insignificant, and as in the recognition test of the previous section, the ratio of the improvement in recognition performance is exceedingly small even when the number of weights is increased by changing the state-clustered HMM into a tied-mixture HMM.

Table 4 shows the processing time and memory requirements of the state-clustered HMM and tied-mixture HMM during the recognition of test set A. As before, the state-clustered HMM has 1,543 weights. The processing time

*Figure 4. Comparison of tied-mixture HMMs with different numbers of weights*

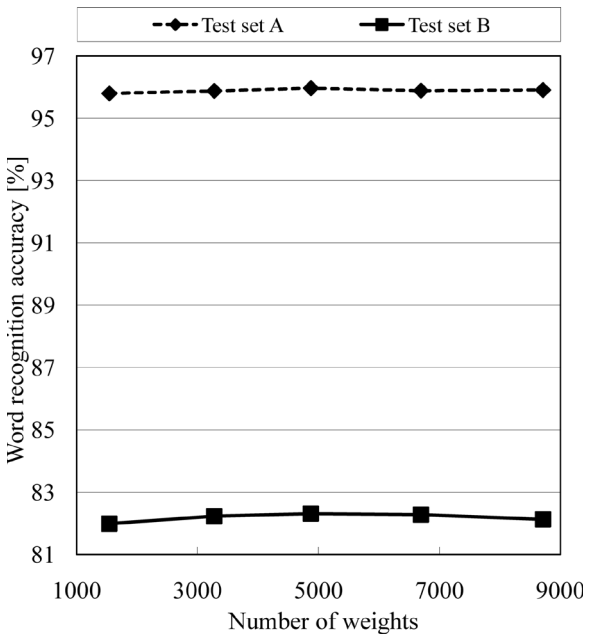


*Table 4. Increase of processing time and memory requirements with number of weights*

| Number of weights | 1,543 | 3,276 | 4,876 | 6,695 | 8,711 |
|---|---|---|---|---|---|
| Number of models | 3,151 | 4,747 | 5,637 | 6,424 | 6,775 |
| Number of nodes | 256,157 | 259,042 | 259,315 | 259,471 | 259,519 |
| Processing time (× real time) | 7.2 | 7.4 | 7.7 | 7.7 | 7.9 |
| HMM computation time (ratio) | 1.00 | 1.05 | 1.08 | 1.10 | 1.12 |
| Memory (MB) | 843 | 869 | 871 | 873 | 875 |

increases as the number of weights increases. Therefore, it can be seen that when a state-clustered HMM is changed into a tied-mixture HMM and the number of weights is increased, it only causes a very small improvement of recognition rates and increase of processing time.

Approximately 70% of the processing time is taken up by HMM computations such as the calculation of output probabilities and the delivery of probability values inside active nodes. It can be seen that the HMM computation time increases as the number of different states in the tied-mixture HMM increases.

Also, the memory requirements of the tied-mixture HMM are greater than those of the state-clustered HMM due to need for more weights and need to store the output probabilities of the increased number of states. However, this increase is very small compared with the amount of memory used in the decoder as a whole.

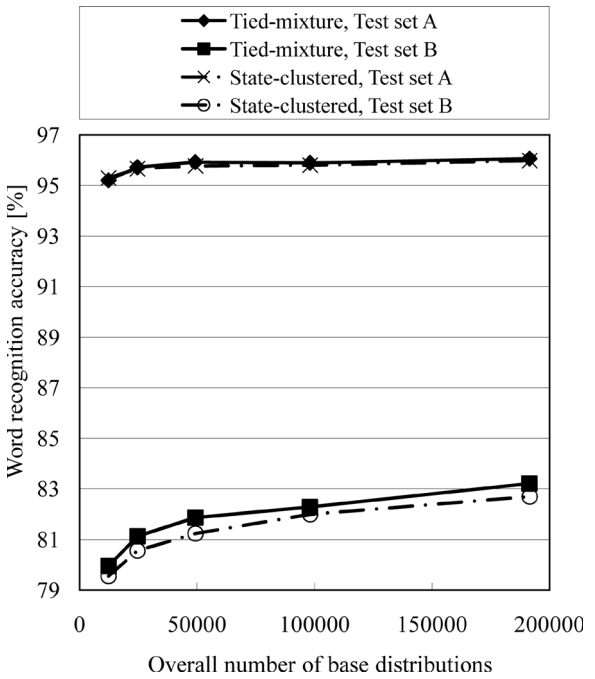
## 5.4. Comparison of HMMs with Different Numbers of Overall Base Distributions

In the recognition tests of section 5.3, increasing the number of weights did not cause a change in recognition rate. It is possible that the quantity of parameters became saturated with respect to the amount of training data. Therefore we set the number of states in the state-clustered HMM and the number of codebooks in the tied-mixture HMM to 1,543, and the number of weights in the tied-mixture HMM to 6,695 and varied the number of base distributions included in the codebooks (using the same number in each HMM). The recognition results are shown in Figure 5.

The word recognition accuracy of the tied-mixture HMM was at most 0.15% better than that of the state-clustered HMM for test set A, and at most 0.62% better for test set B. This difference is insignificant, and as in the previous recognition tests, the ratio of the improvement in recognition performance is exceedingly small even when the number of weights is increased by changing the state-clustered HMM into a tied-mixture HMM.

On comparing state-clustered HMMs in which the overall number of base distributions was 12,000 and 190,000, we found that the word recognition accuracy increased by 0.69% for test set A, and by 3.14% for test set B. Since increasing the number of base distributions results in a better recognition rate, this shows that the training data is not saturated. The reason why increasing the number of base distributions is more effective than increasing the number of weights is because it results in a greater increase in the overall number of parameters. The tied-mixture HMM with 8,711 weights that was used in the recognition tests of section 5.3 only has about 1.05 times as many parameters as the state-clustered HMM with 1,543 weights, whereas the state-clustered HMM of this test, where the overall number of base distributions is 190,000, holds approximately 15 times as many parameters as the state-clustered HMM where the overall number of base distributions is 12,000.

*Figure 5. Comparison of HMMs with different numbers of overall base distributions*




## 5.5. Comparison of HMMs with Different Numbers of Codebooks While the Overall Number of Base Distributions is Kept Constant

We performed recognition tests using state-clustered and tied-mixture HMMs in which the overall number of base distributions was approximately 55,000, while the number of codebooks was varied between 124, 446, 1,756 and 6,981. The overall number of weights in the tied-mixture HMM with 6,981 codebooks was 6,981. Therefore, in the tied-mixture HMM with 6,981 codebooks, states that share the same base distributions must also have the same weights, so this is identical to a state-clustered HMM. The recognition results are shown in Figure 6.

Differences in the recognition performance of state-clustered and tied-mixture HMMs become apparent when the number of codebooks is small. Table 5 shows the difference in word recognition accuracy of a tied-mixture HMM from the state-clustered HMM with the same number of codebooks. When the number of codebooks is small, increasing the number of weights is an effective way of improving the recognition rate, but when the number of codebooks has increased to a certain extent, the ratio of improvement in recognition rate by increasing the number of weights becomes very small.

As in the two recognition tests mentioned above, this is due to differences in the number of parameters between the state-clustered and tied-mixture HMMs. Table 5 also shows the number of parameters for a tied-mixture HMM assuming that for each state-clustered HMM with the same number of codebooks is equal to 1. The tied-mixture HMM where the number of codebooks is 124 has about 1.8 times as many parameters as the state-clustered HMM with the same number of codebooks. On the other hand, the tied-mixture HMM where the number of codebooks is 1,756 has about

*Figure 6. Comparison of HMMs with different numbers of codebooks while the overall number of base distributions is kept constant*

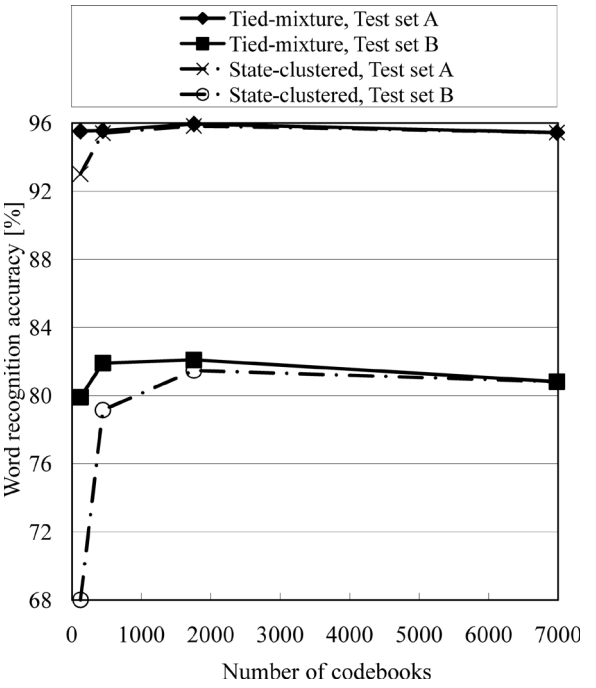



*Table 5. Number of parameters for a tied-mixture HMM and difference for a tied-mixture HMM from each state-clustered HMM*

| Number of codebooks | 124 | 446 | 1,756 | 6,981 |
|---|---|---|---|---|
| Number of parameters (ratio) | 1.80 | 1.20 | 1.05 | 1.00 |
| Difference in word recognition accuracy for test set A (%) | +2.51 | +0.14 | +0.11 | 0 |
| Difference in word recognition accuracy for test set B (%) | +11.89 | +2.75 | +0.63 | 0 |

1.05 times as many as a state-clustered HMM with the same number of codebooks, so as the number of codebooks increases, the difference in the number of parameters between the state-clustered and tied-mixture HMMs disappears and the ratio of improvement in the recognition rate also decreases.

In Sankar (1998), since comparisons are made under conditions where the number of codebooks is small, the total number of parameters ends up being larger for the tied-mixture HMM even when the overall number of base distributions is the same, and this is thought to be one reason why the recognition performance of the tied-mixture HMM was improved.

## 6. CONCLUSION

We have performed a comparison of state-clustered and tied-mixture HMMs, which are

used for the large-vocabulary continuous speech recognition.

We have shown that when the number of codebooks is small, increasing the number of weights by changing a state-clustered HMM into a tied-mixture HMM causes the number of parameters to increase, resulting a better recognition rate. Conversely, we have shown that when the number of codebooks is large, increasing the number of weights causes hardly any change in the number of parameters, so there is no change in the recognition rate.

We have also performed a comparison of HMM training methods, and we have shown that a training method where the LBG algorithm is used to partition the aligned training data into individual clusters, and this partitioned training data is used to perform HMM calculations directly makes it possible to reduce the creation time while achieving the same recognition rate as that of a training method where the EM algorithm is used repeatedly.

In the future, it will be necessary to study methods for creating HMMs that have efficient parameters. In section 5.5, even though a tied-mixture HMM where the number of codebooks is 124 occupies 32 MBytes, we obtained better recognition results with a state-clustered HMM where the number of codebooks is 1,756 and which occupies just 18 MBytes. It therefore seems that the recognition performance does not always improve as the number of parameters is increased. For the efficient creation of HMMs with better recognition performance, it will be necessary to clarify an optimal distribution method for parameters such as the number of base distributions, the number of codebooks, and the number of weights.

## REFERENCES

Ando, A., Imai, T., Kobayashi, A., Homma, S., Goto, J., & Seiyama, N. et al. (2003). Simultaneous subtitling system for broadcast news programs with a speech recognizer. *IEICE Transactions on Information and Systems*, *E86-D*(1), 15–25.

Bahl, L., Jelinek, F., & Mercer, R. (1983). A maximum likelihood approach to continuous speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *PAMI-5*(2), 179–190. doi:10.1109/TPAMI.1983.4767370 PMID:21869099

Clarkson, P., & Rosenfeld, R. (1997). Statistical language modeling using the CMU–Cambridge toolkit. In *Proceedings of the Eurospeech* (pp. 2707-2710).

Delcroix, M., Kinoshita, K., Nakatani, T., Araki, S., Ogawa, A., & Hori, T. et al. (2013). Speech recognition in living rooms: Integrated speech enhancement and recognition system based on spatial, spectral and temporal modeling of sounds. *Computer Speech & Language*, *27*(3), 851–873. doi:10.1016/j.csl.2012.07.006

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series A (General)*, *39*(1), 1–38.

Digalakis, V., & Murveit, H. (1994). High-accuracy large-vocabulary speech recognition using mixture tying and consistency modeling. In *Proceedings of the DARPA Human Language Technology Workshop* (pp. 313-318). doi:10.3115/1075812.1075886

Hofmann, H., Sakti, S., Hori, C., Kashioka, H., Nakamura, S., & Minker, W. (2012). Sequence-based pronunciation variation modeling for spontaneous ASR using a noisy channel approach. *IEICE Transactions on Information and Systems*, *E95-D*(8), 2084–2093. doi:10.1587/transinf.E95.D.2084

Huang, X., Hon, H., Hwang, M., & Lee, K. (1993). A comparative study of discrete, semicontinuous and continuous hidden Markov models. *Computer Speech & Language*, *7*(4), 359–368. doi:10.1006/csla.1993.1019

Hwang, M., Huang, X., & Alleva, F. (1996). Predicting unseen triphones with senones. *IEEE Transactions on Speech and Audio Processing*, *4*(6), 412–419. doi:10.1109/89.544526

Imai, T., Onoe, K., Kobayashi, A., & Ando, A. (2000). Decoder for Japanese broadcast news transcription. *The Journal of the Acoustical Society of Japan (E)*, *21*(1), 29–31. doi:10.1250/ast.21.29

Iso, K., Watanabe, T., & Kuwahara, H. (1988). Design of a Japanese sentence list for a speech database. In *Proceedings of the Spring Meeting of the Acoustical Society of Japan*, (pp. 89-90).

Itou, K., Yamamoto, M., Takeda, K., Takeda, T., Takezawa, T., & Matsuoka, T. et al. (1998). The design of the newspaper-based Japanese large vocabulary continuous speech recognition corpus. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)* (pp. 3261-3264).

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *35*(3), 400–401. doi:10.1109/TASSP.1987.1165125

Kawahara, T. (2012). Transcription system using automatic speech recognition for the Japanese Parliament (Diet). In *Proceedings of the Twenty-Fourth Innovative Applications of Artificial Intelligence Conference* (pp. 2224-2228).

Kobayashi, A., Fujita, Y., Oku, T., Sato, S., Homma, S., Arai, T., & Imai, T. (2013). Live closed-captioning system using hybrid automatic speech recognition for broadcast news. In *Proceedings of the National Association of Broadcasters Broadcast Engineering Conference (NAB BEC)* (pp. 277-283).

Kobayashi, A., Onoe, K., Imai, T., & Ando, A. (1998). Time dependent language model for broadcast news transcription and its post-correction. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)* (pp. 2435-2438).

Lee, A., Kawahara, T., Takeda, K., & Shikano, K. (2000). A new phonetic tied-mixture model for efficient decoding. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 1269-1272). doi:10.1109/ICASSP.2000.861808

Linde, Y., Buzo, A., & Gray, R. (1980). An algorithm for vector quantizer design. *IEEE Transactions on Communications*, *COM-28*(1), 84–95. doi:10.1109/TCOM.1980.1094577

Liu, X., Hieronymus, J., Gales, M., & Woodland, P. (2013). Syllable language models for Mandarin speech recognition: Exploiting character language models. *The Journal of the Acoustical Society of America*, *133*(1), 519–528. doi:10.1121/1.4768800 PMID:23297923

Maekawa, K., Koiso, H., Furui, S., & Isahara, H. (2000). Spontaneous speech corpus of Japanese. In *Proceedings of the Second International Conference of Language Resources and Evaluation (LREC)* (pp. 947-952).

Ney, H., Haeb-Umbach, R., Tran, B., & Oerder, M. (1992). Improvements in beam search for 10000-word continuous speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 9-12). doi:10.1109/ICASSP.1992.225985

Nguyen, L., Anastasakos, T., Kubala, F., LaPre, C., Makhoul, J., & Schwartz, R. et al. (1995). The 1994 BBN/BYBLOS speech recognition system. In *Proceedings of the ARPA Spoken Language Systems Technology Workshop*, (pp. 77-81).

Ogawa, A., Hori, T., & Nakamura, A. (2012). Error type classification and word accuracy estimation using alignment features from word confusion network. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 4925-4928). doi:10.1109/ICASSP.2012.6289024

Onoe, K., Segi, H., Kobayakawa, T., Sato, S., Homma, S., Imai, T., & Ando, A. (2003). Filter bank subtraction for robust speech recognition. *IEICE Transactions on Information and Systems*, *E86-D*(3), 483–488.

Povey, D., & Woodland, P. (2002). Minimum phone error and I-smoothing for improved discriminative training. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 105-108). doi:10.1109/ICASSP.2002.1005687

Sankar, A. (1998). A new look at HMM parameter tying for large vocabulary speech recognition. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)* (pp. 2219-2222).

Schwartz, R., & Austin, S. (1991). A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 701-704). doi:10.1109/ICASSP.1991.150436

Segi, H., Onoe, K., Sato, S., Imai, T., & Ando, A. (1999). An examination of decision tree and the number of models on state tying triphone HMM. In *Proceedings of the Autumn Meeting of the Acoustical Society of Japan* (pp. 123-124).

Segi, H., Tako, R., Seiyama, N., & Takagi, T. (2010). Development of a prototype data-broadcast receiver with a high-quality voice synthesizer. *IEEE Transactions on Consumer Electronics*, *56*(1), 169–174. doi:10.1109/TCE.2010.5439141

Singh, R., Kumatani, K., McDonough, J., & Liu, C. (2012). A signal-separation-based array postfilter for distant speech recognition. In *Proceedings of the Annual Conference of the International Speech Communication Association (INTERSPEECH)*.

Siu, M., Lang, O., Gish, H., Lowe, S., Chan, A., & Kimball, O. (2012). MLLR transforms of self-organized units as features in speaker recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, (pp. 4385-4388). doi:10.1109/ICASSP.2012.6288891

Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, *IT-13*(2), 260–269. doi:10.1109/TIT.1967.1054010

Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., & Liu, X. et al. (2009). *The HTK book*. Cambridge, UK: Cambridge University Engineering Department.

Young, S., Odell, J., & Woodland, P. (1994). Tree-based state tying for high accuracy acoustics modeling. In *Proceedings of the ARPA Human Language Technology Workshop* (pp. 307-312). doi:10.3115/1075812.1075885

*Hiroyuki Segi received the B.S. and M.S. degrees in Physics and the Ph. D. degree in School of Science for Open and Environmental Systems from Keio University, Yokohama, Japan in 1994, 1996 and 2012, respectively. He joined Japan Broadcasting Corporation (NHK) in 1996. Since 1998, he has been with NHK Science and Technology Research Laboratories, Tokyo, Japan. He is a Principal Research Engineer and has been engaged in research on speech synthesis and speech recognition. In 2009, he was a Visiting Researcher at Carnegie Mellon University, Pittsburgh, PA. He is a senior member of IEEE and a member of the Acoustical Society of Japan (ASJ) and the Institute of Electronics, Information and Communication Engineers (IEICE). He has been the recipient of numerous awards including the 54th Motion Picture and Television Engineering Society of Japan (MPTE) Award in 2001, the Excellent Paper Award from IEICE in 2002, the Technology Development Award from ASJ in 2002 and 2011, Hisoka Maejima Award from Teishin Association, Yagami Alumni Award from Keio University Science and Technology in 2012 and the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology in 2013.*

*Kazuo Onoe received the B.E. degree in eletrical engineering in 1993 from Waseda University. He joined Japan Broadcasting Corporation (NHK) in 1993. Since 1997, he has been with the Science and Technology Research Laboratories where he has been engaged in speech recognition research.*

*Shoei Sato received a B.E. degree and a M.E. degree in 1993 from Tohoku University. He also received the Dr. Eng. degree from Waseda University in 2008. Since 1995 he has been with NHK Science and Technology Research Laboratories and engaged in the research on digital satellite broadcasting system. He is currently engaged in automatic speech recognition research.*

*Akio Kobayashi received his B.E. degree from Waseda University and Ph.D. degree from Toyohashi University of Technology. He joined NHK (Japan Broadcasting Corporation) Science and Technology Research Laboratories in 1996. He is currently engaged in speech processing research involving speech recognition. He is a member of the Acoustic Society of Japan (ASJ), the Information Processing Society of Japan, and the Association of Natural Language Processing of Japan.*

*Akio Ando received the B.S. and M.S. degrees from Kyushu Institute of Design, Fukuoka, Japan, in 1978 and 1980, respectively, and the Dr. Eng. degree from Toyohashi University of Technology, Toyohashi, Japan, in 2001. In 1980, he joined the Japan Broadcasting Corporation (NHK). He has been with the NHK Science and Technology Research Laboratories, Tokyo, Japan, since August 1983. He was in charge of developing simultaneous subtitling systems for live broadcast TV programs using speech recognition. Since 2002, he has been engaged in research on audio and acoustics including acoustic signal processing, electroacoustical transducers, cognitive science of acoustics, and spatial sound reproduction. From 2004 to 2006, he was the Director of the Acoustics and Audio Signal Processing Division, and currently he is a Senior Research Engineer of the Advanced Television Systems Research Division. Since 2010, he has been a Guest Professor at the Tokyo Institute of Technology, Japan.*