PAPER Special Section on Robust Speech Processing in Realistic Environments

Noisy Speech Recognition Based on Integration/Selection of Multiple Noise Suppression Methods Using Noise GMMs

Norihide KITAOKA[†a)], ***Member***, **Souta HAMAGUCHI**[††], ***Nonmember***, ***and*** **Seiichi NAKAGAWA**[††], ***Fellow***

SUMMARY To achieve high recognition performance for a wide variety of noise and for a wide range of signal-to-noise ratio, this paper presents methods for integration of four noise reduction algorithms: spectral subtraction with smoothing of time direction, temporal domain SVD-based speech enhancement, GMM-based speech estimation and KLT-based comb-filtering. In this paper, we proposed two types of combination methods of noise suppression algorithms: selection of front-end processor and combination of results from multiple recognition processes. Recognition results on the CENSREC-1 task showed the effectiveness of our proposed methods.

key words: *noisy speech recognition, noise suppression method selection, CENSREC-1*

1. Introduction

In recent years, the performance of automatic speech recognition has been improved drastically by applying statistical approaches. However, most speech recognizers still have the serious problem that their recognition performance degrades in noisy environments. It is necessary to realize robust speech recognition under noisy environments for the improvement of recognition accuracy of systems. A variety of noise suppression methods have been proposed as a front-end of speech recognition. The effect of these methods greatly depends on the noise condition.

There are strong and weak points by the kind and SNR of the noise. In general, it is thought that there are no methods which can effectively suppress various noises over a wide range of SNRs. Therefore, it may be effective to select an appropriate method to each noise condition. This selection seems to be achieved by the noise environment detection.

Some kinds of detection/identification of acoustical events or environments using Gaussian Mixture Models (GMM) have been investigated. In this paper, we propose a method to select an appropriate noise suppression method using GMM for each speech input. The use of GMMs in noisy environments has also been investigated. Lee et al. applied a GMM-based verification of noise and voice signals in a dialog system. Nishimura et al. [2] used GMM to recognize the kind of noise to reflect the behavior of a dialog system. Considering such simple and powerful noise environment classification ability, we first propose a method to select a noise suppression method suitable for a certain noise condition based on GMM likelihood [3]. The front-end processor first selects a suppression method, applies the method to input speech, and sends the feature to the back-end recognizer.

Our above-described solution uses only one recognition procedure, but we can combine several models in parallel to achieve better performance. Fiscus proposed a first voting-based multiple recognizer output combination, ROVER [4]. ROVER and its variations [5]–[7] have achieved good word error reduction. Parallel decoding is also implemented on parallel computation systems to keep computational efficiency as single decoding systems [8].

From the viewpoint of noise robust speech recognition, a method for dealing with diversity of noise SNR using Multi-SNR models [10] has been proposed. Matsuda et al. [9] realized the parallel decoding system considering noise environments, in which multiple acoustic models for various noise environments and multiple speech features are used in parallel and decoded on a parallel computational system.

A hypothesis combination method which combines hypotheses generated by multiple recognition systems using feature streams obtained from multiple noise suppression methods [11] has been proposed. But while some of these methods may be very good at suppressing a certain noise, others may not. Furthermore, they involve huge computational cost. In this paper, we incorporate the GMM likelihood to improve the performance considering the individual performance of the methods on the target noise [12]. We also reduce the computational cost while keeping the advantage of a hypothesis combination method.

We used CENSREC-1 (which is also called AURORA-2J [20]) for evaluation of our method. The CENSREC-1 is a Japanese version of AURORA-2 [13], a common evaluation framework for the noisy connected English digit speech recognition task.

2. Noise Environment Detection Based on GMM

In this paper, we present noise suppression techniques for a speech recognizer for mobile equipment which is used under various noise conditions. To address such a difficult situation, we propose to select noise suppression method(s) suitable for the condition of each recognition process. We

Manuscript received July 2, 2007.

Manuscript revised September 18, 2007.

[†] The author is with Nagoya University, Nagoya-shi, 464–8603 Japan.

[††] The authors are with Toyohashi University of Technology, Toyohashi-shi, 441–8580 Japan.

a) E-mail: kitaoka@nagoya-u.jp

DOI: 10.1093/ietisy/e91–d.3.411

use noise GMMs for the selection. In this section, we first introduce an evaluation framework for noisy speech recognition which matches our supposed situation. Then, we give brief summaries of noise suppression methods which are the choices of our noise selection method and the possibility of the performance improvement by our method using this framework and the individual methods. Finally, we explain the training procedure of noise GMMs.

2.1 Evaluation Framework

We used the CENSREC-1 (AURORA-2J [20]) for evaluation of our method. This evaluation framework includes various noise, some known and some unknown. This framework was constructed considering the same situation we suppose.

The CENSREC-1 is a Japanese version of AURORA-2 [13], a common evaluation framework for the noisy connected English digit speech recognition task. Two training conditions (clean condition/multi-condition) and three testing sets (sets A/B/C) are defined by the CENSREC-1. The sampling rate is 8 kHz. The training data consists of 8440 utterances. The clean-condition training has acoustic models trained by clean speech only. Because a clean speech is not contaminated with noise, the noise suppression methods are not applied to clean training data. The multi-condition training has models trained by a corpus consisting of both clean and noisy speech. In the multi-training set, speech data is contaminated with four kinds of noises (subway, babble, car, exhibition) at every SNR in five variations (clean, 20 dB, 15 dB, 10 dB, 5 dB). The noise suppression methods are applied to the multi-training data as well as the test data. The testing set A includes four different types of noise which were used in the multi-condition training, while the testing set B includes another four different types of noise not used in the multi-condition training. The testing set C then includes noise types from both sets A and B, plus additional convolutional noise. Speech is analyzed using 25 ms frames with a shift of 10 ms. Each word-based HMM had 18 states and 20 Gaussian mixtures per state. The feature vectors consist of MFCC features, energy, their delta and their acceleration (MFCC_E_D_A) of dimension 39.

Relative performance is defined in the CENSREC-1 framework using the accuracy of the target method X_m and the accuracy of the baseline X_b (that is, without suppression), respectively, as follows:

$$\mathit{Relative\ performance} = \frac{X_m - X_b}{100.0 - X_b} \times 100. \quad [\%] \qquad (1)$$

In the original setting of CENSREC-1, Cepstral Mean Normalization (CMN) is not used, and thus the baseline results shown in the following sections are obtained without CMN. But it is well known that the CMN is very effective, so we use it in any other conditions after applying noise suppression methods.

Table 1 Summary of noise suppression algorithms used in the experiments.

	SS	SVD	GMM	KLT
Feature domain	Power spectrum	Waveform	Log Mel-filterbank	Waveform
Assumption about noises	Stationarity	Whiteness	Non-stationarity	Whiteness

In CENSREC-1, evaluation categories were designed to show how much the user's method modified the baseline system except for the change of the front-end process, for example, the adaptation process, model topology, and decoder. Users have to declare the category to which their methods belong. Our methods proposed in this paper belong to the following categories:

Category 0. No changes to the back-end HTK scripts. Changes to only front-end processing, i.e. to feature vectors, can be included in this category.

Category 5. Any process with any computational cost will be allowed.

2.2 Noise Suppression Methods and the Potential of Method Selection

In this paper, as noise suppression methods, spectral subtraction with smoothing of time direction (SS) [14], the temporal domain SVD-based speech enhancement (SVD) [16], GMM-based speech estimation (GMM) [16], [18], and pitch synchronous KLT (KLT) [19] are used[†]. The above four methods are used individually, or combined sequentially: a single method is applied to the input speech and the same or another method is also applied. Sequential uses are denoted as, for example, SS-GMM. Thus, there are totally 21 varieties of noise suppression method including 4 single methods, 16 sequential combinations, and the case without noise suppression.

The algorithms are explained below, and the feature domains in which the compensation is performed and the assumptions about the noises are summarized in Table 1.

Spectral subtraction with smoothing of time direction [14]

The observation signal x is assumed to be the sum of speech signal s and noise n, namely, $x = s + n$. Spectral subtraction [15] in the power spectral domain is defined as below:

$$|\tilde{S}_i(t)|^2 = |X_i(t)|^2 - \alpha|\tilde{N}_i|^2, \qquad (2)$$

where $|\tilde{S}_i(t)|^2$ and $|X_i(t)|^2$ are the i-th components of the estimated power spectrum of speech and the power spectrum of observed signals at the time t, respectively, while $|\tilde{N}_i|^2$ is the i-th component of *a priori* estimated power spectrum of noise, and α is the overestimation factor. We can express $|X_i(t)|^2$ as:

$$|X_i(t)|^2 = |S_i(t)|^2 + |N_i(t)|^2 + 2|S_i(t)||N_i(t)|\cos\theta_i(t), \qquad (3)$$

[†] We selected these four methods becaus we could obtain the software for the respective methods. Thus, the selection of these particular methods has no special meaning. We can adopt arbitrary methods for our selection/integration methods.

where $|S_i(t)|$ and $|N_i(t)|$ are the true values for speech and noise, and $\theta_i(t)$ is the phase difference between speech and noise. We suppose that the speech and the noise do not correlate with each other. The definition of Eq. (2) rests on the fact that the expectation value of $\cos\theta_i(t)$ in Eq. (3) equals zero. However, considering $\cos\theta_i(t)$ as a random variable ranging -1 to 1 and assuming that $\theta_i(t)$ distributes uniformly, the probability density function (pdf) of $\phi = \cos\theta_i(t)$ becomes $f(\phi) = 1/(\pi\sqrt{1-\phi^2})$, a concave function with sole minimum at ϕ=0. Therefore, the term including $\cos\theta_i(t)$ in Eq. (3) cannot be removed even if the noise power can be accurately estimated.

Here, we define the smoothing method as follows [14]:

$$\overline{|X_i(t)|^2} = \sum_r \beta_r |X_i(t-\tau)|^2, \tag{4}$$

where τ = 0,1,...,T-1, $\sum_r \beta_r = 1$. Using (2) and (3), it becomes:

$$\begin{aligned}\overline{|X_i(t)|^2} = \sum_r \beta_r \{&|S_i(t-\tau)|^2 + |N_i(t-\tau)|^2 \\ &+ 2|S_i(t-\tau)||N_i(t-\tau)|\cos\theta_i(t-\tau)|\}.\end{aligned} \tag{5}$$

Assuming that phase differences between speech and noise of successive frames do not correlate with one another, the pdfs of ϕ has the peak at zero and the variance of this term becomes smaller than the original one. Thus, we can assume the third term of (5) is almost zero, and (5) becomes

$$\overline{|X_i(t)|^2} \approx |S_i(t)|^2 + |N_i(t)|^2. \tag{6}$$

Replacing $|X_i(t)|^2$ in (2) with $\overline{|X_i(t)|^2}$, (2) becomes

$$|\tilde{S}_i(t)|^2 \approx |S_i(t)|^2 + |N_i(t)|^2 - \alpha|\bar{N}_i|^2. \tag{7}$$

Therefore, we can estimate the speech signal more accurately if we can estimate $|\tilde{N}_i|$ accurately.

In our experiments in Sect. 5, we used $T = 3$, $\beta_{tau} = 1/3$ for $t = 0, 1, 2$, and $\alpha = 1.8$.

Temporal domain SVD-based speech enhancement [16]

At the i-th windowed short time frame, the observed noisy speech signal $x_i(t)$ is assumed to consist of a clean speech signal, $s_i(t)$, and an additive noise, $n_i(t)$, as follows:

$$x_i(t) = s_i(t) + n_i(t). \tag{8}$$

Therefore, (8) can be represented as (9) in terms of $N \times M$ Toeplitz matrices where N and $M-1$ are an interval length and a maximum delay, respectively:

$$X_i = S_i + N_i. \tag{9}$$

By applying SVD to X_i, X_i is decomposed into three matrices and reconstructed as $X_i = U_i \Sigma_i V_i^T$. As a result, a singular value matrix is obtained. Here, the singular value can be represented as (10) under the assumption that $s_i(t)$ is not correlated with $n_i(t)$:

$$\sigma_m^{X_i} = \sigma_m^{S_i} + \sigma_m^{N_i}, \tag{10}$$

where $m = 0, \ldots, M-1$. In (10), if $n_i(t)$ is white noise, it can be assumed that the distribution of $\sigma_m^{N_i}$ is uniform. Therefore, $\sigma_m^{S_i}$ can be estimated as (11):

$$\hat{\sigma}_m^{S_i} = \sigma_m^{X_i} - \bar{\sigma}^{N_i}. \tag{11}$$

By using estimated $\hat{\sigma}^{S_i}$, the Toeplitz matrix $\hat{S}_i$ is estimated as in the following [17]:

$$\hat{S}_i = U_i W_i \Sigma_i V_i^T, \tag{12}$$

$$W_i = diag\left(\frac{\sigma_m^{X_i} - \bar{\sigma}^{N_i}}{\sigma_m^{X_i}}\right). \tag{13}$$

In (10), if it can be assumed that the singular values of clean speech $\sigma_m^{S_i}$ vanish for a sufficiently large index of $m(m \geq R)$, the remaining singular values can be handled as singular values of the noise. Thus:

$$\sigma_m^{N_i} \quad \cong \quad \sigma_m^{X_i} \quad (m \geq R). \tag{14}$$

From this fact, the averaged singular value is estimated as:

$$\bar{\sigma}^{N_i} = \frac{1}{M-R}\sum_{m=R}^{M-1} \sigma_m^{X_i}. \tag{15}$$

In Sect. 5, we used $M = 28$ and $N = 173$, and R was set as if the cumulative contribution rate up to R-th singular value was beyond 90%.

GMM-based speech estimation [16], [18]

At the i-th frame, the logarithmic output energy of a Mel filter bank of observed noisy speech is represented as follows:

$$\begin{aligned}X_{\log}(i) &= \log[\exp(S_{\log}(i)) + \exp(N_{\log}(i))] \\ &= S_{\log}(i) + \log[1 + \exp(N_{\log}(i) - S_{\log}(i))] \\ &= S_{\log}(i) + G_{\log}(i),\end{aligned} \tag{16}$$

$$G_{\log}(i) = \log[1 + \exp(N_{\log}(i) - S_{\log}(i))], \tag{17}$$

where $X_{\log}(i), S_{\log}(i)$ and $N(i)$ denote the vectors that have logarithmic output energy of a Mel filter bank of observed noisy speech, clean speech and noise, respectively. In (16), $G_{\log}(i)$ is equivalent to the mismatch factor between $X_{\log}(i)$ and $S_{\log}(i)$.

First, suppose that $S_{\log}(i)$ can be modeled by GMM with K mixture distributions,

$$p(S_{\log}(i)) = \sum_{k=1}^{K} P(k) N(S_{\log}(i); \mu_{S,k}, \sigma_{S,k}), \tag{18}$$

where $p(S_{\log}(i))$ denotes the output probability of $S_{\log}(i)$, and $P(k)$, $\mu_{S,k}$ and $\sigma_{S,k}$ denote the mixture distributions as well as $S_{\log}(i)$. When GMM of $S_{\log}(i)$ is given, $X_{\log}(i)$ is also represented by GMM using the following description. Let μ_N denote the mean vector of $N_{\log}(i)$, which is estimated using the first 10 frames of the observed noisy speech, $X_{\log}(i)$. The mean vector of $X_{\log}(i)$ at the k-th Gaussian distribution is then estimated as follows based on (16) and (17):

$$\mu_{X,k} \simeq \mu_{S,k} + \log[1 + \exp(\mu_N - \mu_{S,k})] = \mu_{S,k} + \mu_{G,k}. \tag{19}$$

On the other hand, the covariance matrix of $X_{\log}(i)$ can be estimated as (20):

$$\Sigma_{X,k} \simeq \Sigma_{S,k}. \tag{20}$$

In (19), $\mu_{G,k}$ corresponds to the mean vector of the mismatch factor at k-th Gaussian distribution. Therefore, the expectation of $G_{\log}(i)$ is estimated as the weighted average of $\mu_{G,k}$ using a posterior probability $P(k|X_{\log}(i))$ as follows:

$$\hat{G}_{\log}(i) = \sum_{k=1}^{K} P(k|X_{\log}(i))\mu_{G,k}, \tag{21}$$

$$P(k|X(i)) = \frac{P(k)N(X_{\log}(i);\mu_{X,k},\Sigma_{X,k})}{\sum_{k'=1}^{K} P(k')N(X(i);\mu_{X,k'},\Sigma_{X,k'})}. \tag{22}$$

From the above-described procedure, the clean speech $\hat{S}_{\log}(i)$ is estimated by subtracting $\hat{G}_{\log}(i)$ from $X_{\log}(i)$ as (23):

$$S_{\log}(i) = X_{\log}(i) - \hat{G}_{\log}(i). \tag{23}$$

Here, we set $K = 512$.

KLT-based comb-filtering [19]

In KLT-based comb-filtering, each sample of the clean speech $s(t)$ of the t-th frame is reconstructed from the estimation of $(2T + 1)$ dimensional vectors $S_p(t, i)$ at the t-th frame:

$$\begin{aligned} S_p(t,i) = (&s((t-T-1)K+i),\\ &\ldots, s((t+T-1)K+i))^T, \end{aligned} \tag{24}$$

where i is from 1 to L, which is the frame length, K is the pitch period, and T is set to 3 in the experiments in Sect. 5 Assuming that noise is additive, we have the noisy input signal:

$$X_p(t,i) = S_p(t,i) + N_p(t,i), \tag{25}$$

where $N_p(t,i)$ is a $(2T + 1)$ dimensional noise vector. Now, let H be a $(2T + 1) \times (2T + 1)$ linear estimator of the clean speech vector as follows:

$$\hat{S}_p = HX_p. \tag{26}$$

The error signal obtained in this estimation is given by

$$r = \hat{S}_p - S_p = (H - I)S_p + HN_p = r_S + r_n, \tag{27}$$

where $r_s = (H - I)S_p$ represents signal distortion and $r_n = HN_p$ represents residual noise. We define the energies of signal distortion $\overline{\varepsilon_S^2}$ and residual noise $\overline{\varepsilon_n^2}$, respectively, as follows:

$$\overline{\varepsilon_S^2} = trE\{r_S r_S^T\} = tr\{(H-I)R_S(H-I)^T\}, \tag{28}$$

$$\overline{\varepsilon_n^2} = trE\{r_n r_n^T\} = tr\{HR_nH^T\}, \tag{29}$$

where R_s and R_n are covariance matrices of the clean signal and the noise vector, respectively. Now, assuming R_s and R_n are provided, the linear estimator is obtained from

$$\min_H \overline{\varepsilon_S^2}, \quad subject\ \ to: \frac{1}{K}\min_H \overline{\varepsilon_n^2} \leq \sigma_n^2, \tag{30}$$

where σ_n^2 is a positive constant. H is a stationary feasible point if it satisfies the gradient equation of the Lagrangian

$$L_H(H,\mu) = \overline{\varepsilon_S^2} + \mu(\overline{\varepsilon_n^2} - K\sigma_n^2), \tag{31}$$

$$\mu(\overline{\varepsilon_n^2} - K\sigma_n^2) = 0 \quad for\ \ \mu \geq 0, \tag{32}$$

where μ is the Lagrange multiplier. From (27), (28), we obtain:

$$H = R_S(R_S + \mu R_n)^{-1}. \tag{33}$$

Now, let the eigenvalue decomposition of R_S be defined as follows:

$$R_S = U\Lambda_S U^T, \tag{34}$$

where Λ_S is a diagonal $(2T+1)\times(2T+1)$ matrix that contains clean signal covariance matrix eigenvalues and U contains its eigenvectors. U is called the inverse KLT and the unitary U^T is called KLT. Substituting (33) into (32), we obtain

$$H = U\Lambda_S(\Lambda_S + \mu U^T R_n U)^{-1}U^T. \tag{35}$$

Assuming that noise is white, we can rewrite the estimator as

$$H = UGU^T, \tag{36}$$

where

$$G = diag(g_t(1), g_t(2), \ldots, g_t(2T+1)), \tag{37}$$

$$g_t(i) = \lambda_S^i/(\lambda_S^i + \mu\lambda_n), \tag{38}$$

where λ_S^i and λ_n are the i-th diagonal component of Λ_S and the variance of the noise, respectively. The signal $\hat{S}_p = HX_p$ is obtained by applying the KLT to the noisy signal.

Yamada et al. [21] showed the effectiveness of the selection algorithms from various noise suppression methods and their combinations heavily depend on noise conditions.

Table 2 shows the word recognition accuracy based on a baseline system, and Table 3 shows the recognition performance based on the manual selection. In this experiment, a noise suppression method is selected for each noise condition (a combination of a kind of noise and SNR) from the 21 variations of the noise suppression methods as shown in Table 4. In this table, SS, SVD, GMM, and KLT are described as S, T, G, and K, respectively, and their sequential uses are described using '-'. A, B, and C express the kind of test sets. The average absolute word accuracy and the relative performance in the clean training and the multi-training are shown in the tables.

Table 3 shows the relative performance of the manual selection in the clean training and the multi-training, and

Table 2 Word accuracy by baseline system (%).

%Acc				
	A	B	C	Overall
Clean Training	46.51	43.98	49.90	46.17
Multicondition training	91.53	80.39	85.83	85.93
Average	69.02	62.18	67.86	66.05

Table 4 Manually selected the best method for each noise condition.

(a) Clean training.

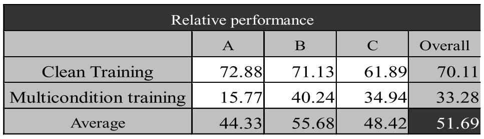
Noise set	A				B				C (with convolutional noise)	
Noise	Subway	Babble	Car	Exhibition	Restaurant	Street	Airport	Station	Subway	Street
clean	G-G	T-G	K-G	G-G	G-G	T-G	K-G	G-G	G-G	G
20 dB	T-K	G-K	K-G	G-K	G-K	K-G	G-K	G-K	G-k	G-K
15 dB	G-K	G-K	T-G	G-K	G-K	G-K	G-K	G-K	G-K	K-G
10 dB	G-K	G-K	K-G	G-K	G-K	K-G	G-K	S-G	T-K	K-G
5 dB	K-K	G-K	S-G	G-K	G-K	K-G	G-K	S-G	T-K	K-G
0 dB	K-K	G-K	S-G	T-G	G-K	K-G	S-G	S-G	K-K	T-G
−5 dB	K-K	G-K	S-G	T-G	G-K	K-K	G-S	S-G	K-K	T-G

(b) Multi-training.

Noise set	A				B				C (with convolutional noise)	
Noise	Subway	Babble	Car	Exhibition	Restaurant	Street	Airport	Station	Subway	Street
clean	S	S-G	T	S-G	S	S-G	T	S-G	S	G-G
20 dB	G-G	G-G	S-G	S	S-T	S-G	S-T	S-T	S	S-G
15 dB	T	S-G	S	S	T-S	G-G	S	T-G	G	G-S
10 dB	G-K	S	S-G	G-G	K-S	K-G	T-S	S-T	G-G	G-G
5 dB	T-K	K-T	S-G	T-G	K-S	K-G	S-S	S-G	T-G	S
0 dB	T-K	S-T	S-T	T-G	K-S	K-G	S-G	S	S	S
−5 dB	K	S-S	S-T	K	S-S	S	S	S-S	T-G	S

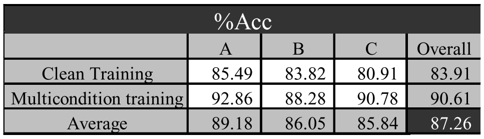
Table 3 Result by selecting the best method for each noise condition (%).

(a) Relative performance.



Relative performance				
	A	B	C	Overall
Clean Training	72.88	71.13	61.89	70.11
Multicondition training	15.77	40.24	34.94	33.28
Average	44.33	55.68	48.42	51.69

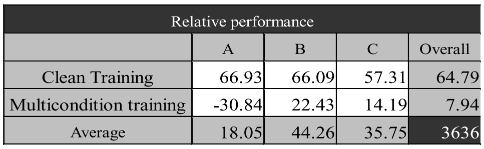
(b) Word accuracy.



%Acc				
	A	B	C	Overall
Clean Training	85.49	83.82	80.91	83.91
Multicondition training	92.86	88.28	90.78	90.61
Average	89.18	86.05	85.84	87.26

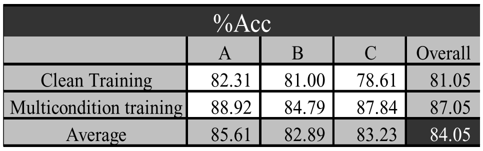
Table 5 Result by GMM-KLT, which achieved the best performance in clean training condition (%).

(a) Relative performance.



Relative performance				
	A	B	C	Overall
Clean Training	66.93	66.09	57.31	64.79
Multicondition training	-30.84	22.43	14.19	7.94
Average	18.05	44.26	35.75	3636

(b) Word accuracy.



%Acc				
	A	B	C	Overall
Clean Training	82.31	81.00	78.61	81.05
Multicondition training	88.92	84.79	87.84	87.05
Average	85.61	82.89	83.23	84.05

Table 5 shows the relative performance of a method (GMM-KLT; the sequential combination of GMM and KLT), whose relative performance was the highest in clean training. Table 6 shows the relative performance of the method (SVD-GMM), whose relative performance was the highest in the multi-training. From comparing these performances as shown in Tables 3, 5, and 6, the best method is selected for each noise condition. Thus, instead of applying a single method for the best performance, an improvement is obtained.

2.3 Noise GMM Training for Automatic Noise Environment Detection

Here, we explain how to train GMMs to evaluate the likelihood of noise environments, which are used to select noise environments in the methods explained in the following sections. The speech data were contaminated with four kinds of noises by five variations of SNRs. Thus there are 20 kinds of noise conditions in the training data[†]. The best suppression method for each noise condition is applied to all the speech under each condition. Table 7 shows the best method for each condition in the multi-training data set. The suppression method applied to noisy speech is selected by using GMMs. Figure 1 shows the procedure of the GMM training. In the experiments, we used the first 10 frames of each speech file in the CENSREC-1 training data as the noise data. We gathered all the noise data of the noise conditions for which a certain suppression method worked best and trained a GMM corresponding to the suppression methods using the noise data. In the recognition stage, the system compared the GMM likelihoods of the noise preceding the speech.

[†] Strictly speaking, the number of noise conditions are 17 = 4 (kinds of noises) * 4 (SNRs) + 1 (clean). In CENSREC-1, the clean data for four kind of noises are different from each other and so the recognition accuracy obtained from the sets is also slightly different. In our experiment, we treated the conditions of these sets as 4 different conditions and thus we had totally 20 conditions. The performance for clean data of these suppression methods, however, was not so different, and so this treatment had little effect on the final results.

Table 6 Result by SVD-GMM, which achieved the best performance in multi-training condition (%).

(a) Relative performance.

Relative performance				
	A	B	C	Overall
Clean Training	63.77	63.73	54.28	61.99
Multicondition training	7.02	29.43	30.94	25.10
Average	28.38	46.58	42.61	41.47

(b) Word accuracy.

%Acc				
	A	B	C	Overall
Clean Training	80.62	79.68	77.10	79.54
Multicondition training	91.93	86.16	90.21	88.88
Average	85.78	82.92	83.65	84.21

Table 7 The best method for each condition under multi-training.

	Subway	Babble	Car	Exhibition
20 dB	GMM	GMM-GMM	SS-GMM	SS
15 dB	SVD	SS-GMM	SS	SS
10 dB	GMM-KLT	SS	SS-GMM	GMM-GMM
5 dB	SVD-KLT	KLT-SVD	SS-GMM	SVD-GMM

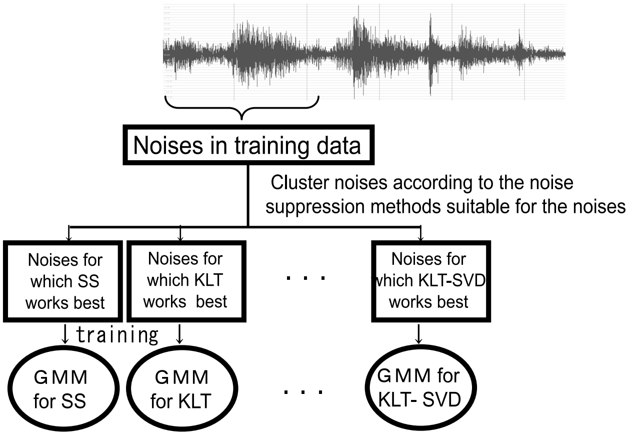

Noises in training data
Cluster noises according to the noise suppression methods suitable for the noises
Noises for which SS works best
Noises for which KLT works best
· · ·
Noises for which KLT-SVD works best
training
GMM for SS
GMM for KLT
· · ·
GMM for KLT- SVD

Fig. 1 Training procedure of GMMs for selecting noise suppression methods including no suppression and the sequential use of the methods.

3. Automatic Selection of Noise Suppression Methods for Front-end Processing

3.1 Speech Recognition Based on Automatic Selection of Noise Suppression Methods

Based on the noise decision, we propose a method of selecting the best noise suppression method in the front-end. After selecting one of the suppression methods corresponding to the GMM with the maximum likelihood. The system applies the method to the input speech and then recognizes it. We used GMMs with 64 diagonal covariance matrices. The first 10 frames of each speech data were used as the noise. Each noise feature consisted of 12 dimensional MFCC and a log energy. The performance of the noise environment detection is 54% when considering the selection correct if the best method for training data is selected. This performance does not seem so good, but the method with a performance tendency similar to the 'best' tends to be selected. And, even if a selection error occurs, a method other than the best one also has some level of recognition performance, and thus the error does not lead to serious degradation.

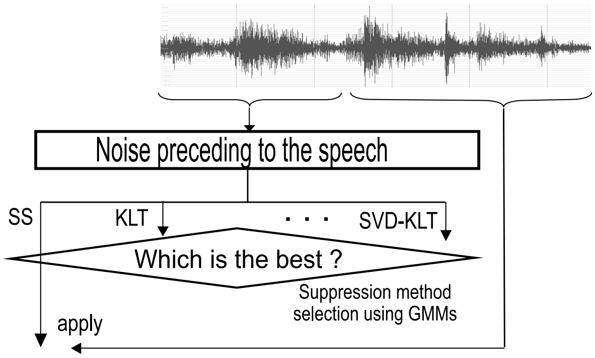

Noise preceding to the speech
SS
KLT
· · ·
SVD-KLT
Which is the best ?
Suppression method selection using GMMs
apply

Fig. 2 Recognition procedure using automatic selection of noise suppression methods.

Figure 2 describes the procedure of the noise suppression using the selection of noise suppression methods. In this figure, SS is selected as the best method by way of example.

The advantage of this method is to be able to select an appropriate suppression method robustly even if the noise is unknown. GMM is trained only using the noises in the training data. Thus, the noise that does not exist in the training data is an unknown noise. We expect that the system selects a method for known noises similar to unknown ones and that the method may be effective for the noise. With this method, the back-end recognizer needs only one HMM set and does not need any special processing. Therefore, this method can be applied to distributed speech recognition.

3.2 Iterative Training of Acoustic Model

The proposed method is for the noise suppression only by the front-end and we do not modify the back end except for acoustic models. In clean training condition, the suppression methods are applied only to the test data. Therefore, there is no modification of the acoustic models even if the front-end applies a different method to each input speech. However, the acoustic models can be retrained using the training data compensated by various suppression methods in the multi-training condition. Retraining tends to improve recognition performance, but the appropriate method for every noise condition (i.e. Table 7) may change because of the retraining. So we select the best suppression method for each noise condition after each iteration and make GMM again (for each noise condition group). Then we can obtain new acoustic models from the training data to which the selected noise suppression method by the new GMMs is applied. We iterate this procedure and stop it when all the correspondences between noise conditions and suppression methods are fixed. Figure 3 shows the procedure of the iterative training.

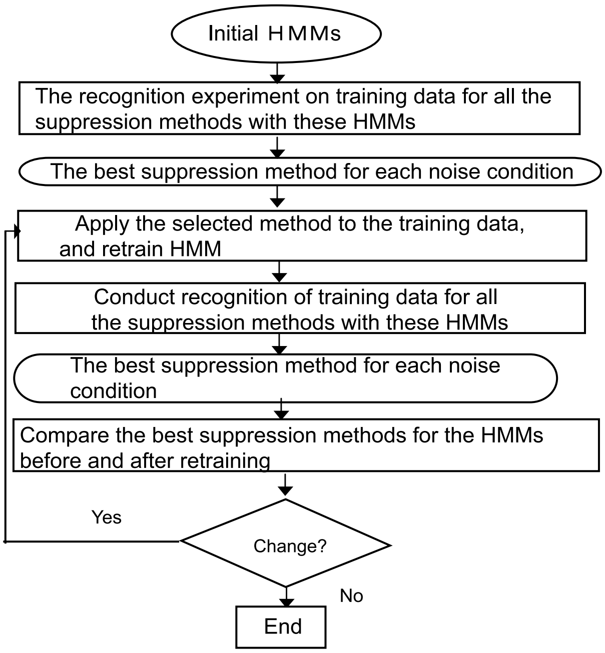

Initial HMMs
The recognition experiment on training data for all the suppression methods with these HMMs
The best suppression method for each noise condition
Apply the selected method to the training data, and retrain HMM
Conduct recognition of training data for all the suppression methods with these HMMs
The best suppression method for each noise condition
Compare the best suppression methods for the HMMs before and after retraining
Yes
Change?
No
End

Fig. 3 Iterative training procedure for acoustic model.

First we use appropriate acoustic models (HMMs) as initial models. We conduct recognition experiments on training data for all the suppression methods with these HMMs, and then select a method with the best accuracy for each noise condition. The GMM for each suppression method is trained according to the recognition result. The recognition experiments are conducted with these HMMs, and the best suppression method for each noise condition is selected again. If all the correspondences between noise conditions and the suppression method are the same as the correspondence just before the last HMM training, the iteration terminates. If not, we conduct the above procedure again.

4. Integration of Recognition Results —Integration in Back-end—

4.1 Front-end Processing vs. Back-end Processing

The integration of the suppression methods in the front-end obtains the accuracy improvement to some degree without increasing computational cost on the back-end processing. On the other hand, the integration of the noise suppression methods in the back-end has been proposed [11]. The integration is done by voting. The recognizer corresponding to each noise suppression method votes for the hypothesis obtained by the recognizer, and the hypothesis which gets majority vote is selected as a final result (voting method). This method showed a significant improvement in accuracy. However, the computational cost was huge. So, we investigate a method to improve the recognition accuracy with less computational cost using the GMM-based selection of noise suppression method.

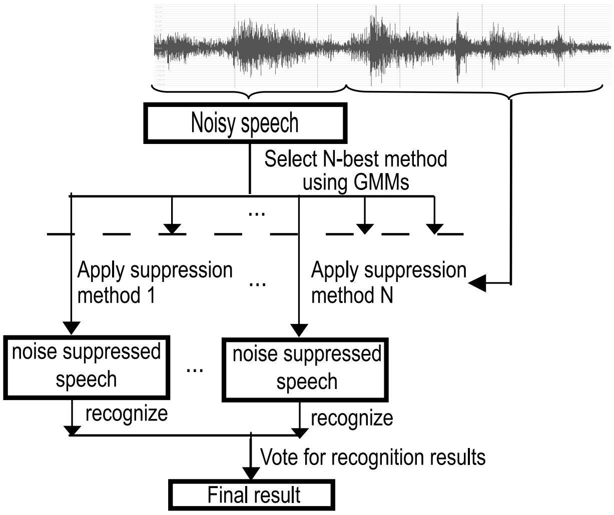

Noisy speech
Select N-best method using GMMs
...
Apply suppression method 1
...
Apply suppression method N
noise suppressed speech
...
noise suppressed speech
recognize
recognize
Vote for recognition results
Final result

Fig. 4 Voting procedure using GMMs.

4.2 Integration of Recognition Results Using GMM

To reduce the computational cost of a voting method, the system first selects some effective suppression methods which are performed in parallel. Then one votes for the results. In this strategy, GMMs are used as the case with the method in the front-end. Figure 4 shows the voting algorithm procedure using GMMs. The training procedure of GMM is similar to Sect. 5. The noise feature is inputted to each GMM, the likelihood of 21 suppression methods is obtained, and the N-best noise suppression methods are selected. Then, recognition procedures using selected noise suppression methods are performed in parallel. The hypotheses obtained from these procedures are voted, and the digit sequence hypothesis with maximum vote is adopted as the final result. When the number of votes is the same for plural hypotheses, or when all the hypotheses are different from each other, the hypothesis generated by the method with the highest likelihood of noise-GMM is adopted. Moreover, because there are differences among the effects of the suppression methods, it is natural to assign priorities to the methods according to the noise conditions. Therefore, we use a weighted voting method based on the likelihood (or priority) of GMMs.

5. Experiment

5.1 Front-end Processing Results

We evaluated the method described in Sect. 4 on the CENSREC-1. The whole noise suppression procedure is done in the front-end, and all methods are categorized as *category 0* [20].

5.1.1 Result of Clean Training

We evaluated three noise suppression methods: GMM-KLT,

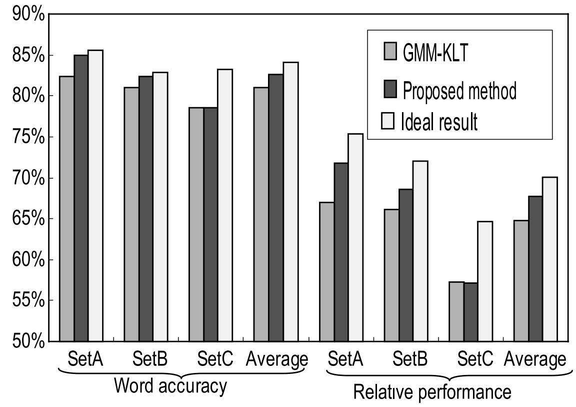


Fig. 5 Performance under clean training condition (%).

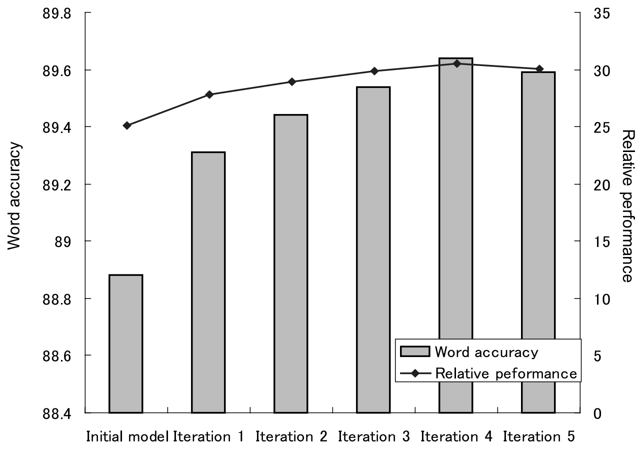


Fig. 6 Performance improvement by iterative training (%).

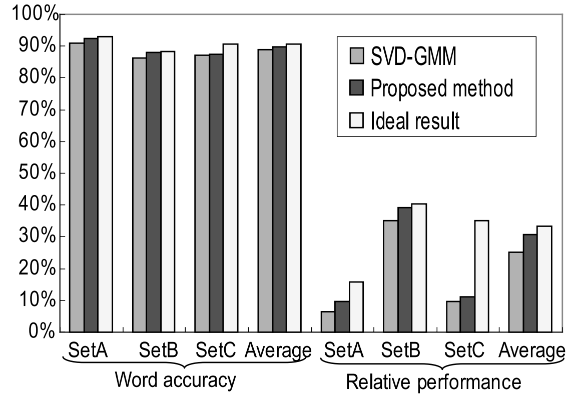


Fig. 7 Recognition performance of multi-condition training (%).

which was the best single (sequential) suppression method among all under clean training condition, the proposed method, and the manual selection of the best suppression method for each noise condition (ideal result). The selection accuracy of noise suppression methods by GMMs was about 54% both under clean and multi-conditions. Figure 5 shows the results in word accuracy and improvement relative to the baseline.

The proposed method obtained the relative performance improvement of 67.7% as compared to the baseline, which was better than “GMM-KLT” (64.8%). That is to say, we could obtain better performance with the proposed method than all the individual methods included in the selection of the proposed method. The improvement of the recognition accuracy of test set B (speech contaminated with unknown noises) is not so inferior to the improvement of the recognition accuracy of test set A (with known noise). This proved that our proposed method could suppress not only known but also unknown noises robustly.

Unfortunately, our method could not improve the accuracy of test set C to which convolutional noise was added. The method to compensate for channel distortion such as CMN was not applied to the features for evaluating noise GMMs, and thus the noise environment selection did not work well on the Set C.

5.1.2 Result of Multi-Training

We evaluated three noise suppression methods: SVD-GMM, which was the best single (sequential) suppression method among all under multi-training condition, the proposed method, and the manual selection of the best suppression method for each noise condition (ideal result). We used the HMMs trained from the speech applied with “SVD-GMM,” which is the best combination method for multi-condition training among the 21 methods. We used the GMM obtained by the training method described in Sect. 3.2. Figure 6 shows the result of the acoustic models obtained at every iteration step. The bars and the lines show the word accuracies and the relative performance improvement to the baseline, respectively. The changes of the correspondences between noise conditions and the suppression method decreases as the process were iterated, and no change occurred after the fifth iteration. The best performance was obtained by the HMMs trained at the fourth iteration, after which we obtained the absolute word accuracy improvement of 0.2%. The word accuracy was 85.93% when the noise was not suppressed (baseline). Figure 7 shows the recognition results for the SVD-GMM, the proposed method, and the ideal method. The proposed method obtained the relative performance improvement of 30.5% as compared to the baseline. Compared with the SVD-GMM, the improvement of relative improvement was 5.4% from “SVD-GMM” (30.5% from 25.1%). Hence, the proposed method could obtain better relative performance than all the individual methods. This method worked well even for unknown noises from the result on test set B.

The noise environment detection rate of noise GMM was about 54%. The ideal result assumed that the detection rate was 100%, and thus the difference of this detection performance results in a difference in the recognition performance of the ideal and proposed methods in Figs. 5 and 7.

5.2 Integration in Back-end Processing

We evaluated the integration method in the back-end. In this method, we modified the back-end processing and thus this method is categorized as *category 5* [20]. The advantage of this method is that it allows one to use noise suppression method-dependent HMMs, so we evaluated this method under the multi-training condition.

In our method, the noise suppression methods were dynamically selected on the fly. For comparison, we also conducted the voting by fixed N methods. These N methods were selected *a priori* by overall recognition performance on the training data. We conducted the recognition experiment by the voting method with N noise suppression methods with N = 1, 5, 10, 15 and 21 (used all suppression methods) on the multi-condition training. We could use multiple hypotheses for voting, so we tested the 1-best and 5-best hypotheses per noise suppression method. Figure 8 shows the results. In Fig. 8 'baseline' describes the method with the fixed N noise suppression methods and 'proposed' describes the method with dynamically selected N noise suppression methods. The recognition accuracy of the proposed method was higher than that of baseline. Because all methods were used, the recognition accuracy was the same when using $N = 21$ for both voting methods. When using $N = 1$, 'baseline' was the best single method, and 'proposed' selected a suitable method for every noise condition by using GMM. We found the absolute improvement of 0.34% (2.4% relative) when using $N = 5$.

All the accuracy was slightly improved using the 5-best hypotheses for voting and we observed almost same tendency as was in the case of 1-best. Utterance-wise selection works well, and thus the performance of our integration method with $N = 5$ was superior to the manual selection for each noise condition shown in Table 3.

Table 8 shows the results in word accuracy and string accuracy. We tested the improvement of the method with 5 dynamically selected methods from the fixed 5 methods in string accuracy using sign test and proved that there was a significant improvement with the significance level of 1%.

This method requires a computational cost almost directly proportional to the number of selected methods, and so Fig. 8 shows the relation between computational cost and performance. In light of the two lines indicating 1-best results in Fig. 8, dynamic noise suppression method selection obtains comparable performance at less computational cost than using fixed N noise suppression methods. For example, almost the same performance as 10 noise suppression methods can be obtained by dynamic 7 or 8 method selection. This means a 20–30% cost reduction. It does not cost much to obtain N-best candidates from each recognition process. From the viewpoint of computational cost, the recognition performance reached saturation at about ten times the computational cost. Adopting the N-best candidate slightly improves the performance without increasing the computational cost. Recently, parallel decoding on multiple processors has been proposed [8], [9]. Our method takes almost the same process time as conventional single decoding when using such implementation. Also, our method can balance the computational cost and recognition performance by controlling the number of selected methods according to the number of available processors.

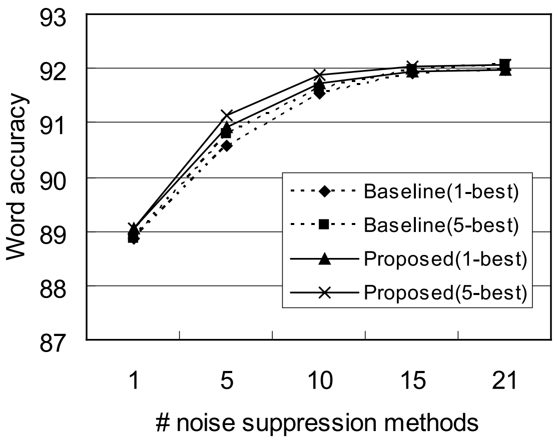


Fig. 8 Recognition accuracy for voting method with N noise suppression methods (%). Comparison between the method with fixed N methods and the method with dynamically selected N method.

Table 8 Comparison of proposed methods and baseline.

	Word acc.	String acc.
Fixed 5 methods	90.79%	81.23%
Dynamically selected 5 methods	91.15%	82.41%
Voting without weight (21 methods)	91.97%	84.38%
Weighted voting by GMM (21 methods)	92.20%	84.60%

We also evaluated the weighted voting method. We used 1.5 and 0.5 as the weights for the 1/3 of suppression methods with high likelihoods of noise GMMs and for the 1/3 with low likelihoods, respectively. Results are shown in Table 8, and we proved that a significant improvement was achieved with the weighted voting method with the significance level of 1% by a sign test [22]. We obtained the word accuracy improvement of 0.23% (the relative performance improvement of 1.63%) and the string accuracy improvement of 0.22% by the voting with weight.

6. Conclusion

We proposed the automatic selection of noise suppression method using GMM corresponding to each noise suppression method. We also proposed an iterative training of HMMs and GMMs for multi-conditional training. We first proposed to apply the method selection to the front-end processing. We evaluated the proposed method using CENSREC-1 Japanese noisy connected digit speech recognition task and obtained better recognition performance than all the individual methods including the sequential combinations in both clean and multi-training. Then, we proposed the integration method in which noise suppression methods were dynamically selected using GMM in back-end. We found the absolute improvement of 0.36% as compared to the method with fixed N noise suppression methods when using $N = 5$ with 5-best hypotheses per suppression methods.

We proved that our method could manage multiple noise suppression methods efficiently to complement each other. Our method, of course, can adopt other suppression methods to achieve further improvement.

Acknowledgments

The authors thank Prof. Kazuya Takeda of Nagoya University and Dr. Masakiyo Fujimoto of NTT for their provision of various noise suppression software. We also thank Prof. Takeshi Yamada of University of Tsukuba for his assistance in this study. The presented study was conducted using CENSREC-1/AURORA-2J database developed by IPSJ-SIG SLP Noisy Speech Recognition Evaluation Working Group.

References

[1] A. Lee, K. Nakamura, R. Nisimura, H. Saruwatari, and K. Shikano, "Noise robust real world spoken dialogue system using GMM based rejection of unintended inputs," INTERSPEECH2004-ICSLP, vol.I, pp.173–176, 2004.

[2] R. Nisimura, A. Hashizume, T. Irino, and H. Kawahara, "Human-robot interaction interface using GMM-based noise recognition," WESPAC IX 2006, vol.347, pp.26–28, 2006.

[3] S. Hamaguchi, N. Kitaoka, and S. Nakagawa, "Robust speech recognition under noisy environments based on selection of multiple noise suppression methods," IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP2005), pp.308–313, 2005.

[4] J.G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," Proc. ASRU, pp.247–354, 1997.

[5] H. Schwenk and J-L. Gauvain, "Combining multiple speech recognizers using voting and language model information," Proc. 6th. ICSLP, pp.915–918, 2000.

[6] T. Utsuro, H. Nishizaki, Y. Kodama, and S. Nakagawa, "Estimating highly confident portions based on agreement among outputs of multiple LVCSR models," Systems and Computers in Japan, vol.35, no.7, pp.33–40, 2004.

[7] V. Goel, S. Kumar, and W. Byrne, "Segmental minimum Bayes-risk decoding for automatic speech recognition," IEEE Trans. Speech Audio Process., vol.12, no.3, pp.234–250, 2004.

[8] T. Shinozaki and S. Furui, "Spontaneous speech recognition using a massively parallel decoder," ICSLP-2004, pp.1705–1708, 2004.

[9] S. Matsuda, T. Jitsuhiro, K. Markov, and S. Nakamura, "ATR parallel decoding based speech recognition system robust to noise and speaking styles," IEICE Trans. Inf. & Syst., vol.E89-D, no.3, pp.989–997, March 2006.

[10] M. Ida and S. Nakamura, "HMM composition-based rapid model adaptation using *a priori* noise GMM adaptation evaluation on AURORA2 corpus," Proc. ICSLP2002, pp.437–440, 2002.

[11] J. Okada, T. Yamada, and N. Kitawaki, "Integration of recognition results from multiple noise reduction algorithms," 2004 Spring Meeting of the Acoustical Society of Japan, pp.157–158, 2004.

[12] N. Kitaoka, S. Hamaguchi, and S. Nakagawa, "Noisy speech recognition based on selection of multiple noise suppression methods using noise GMMs," ICSLP-2006, pp.2566–2569, Sept. 2006.

[13] H.G. Hirsh and D. Pearce, "The AURORA experimental frame work for the performance evaluations of speech recognition systems under noisy conditions," ISCA ITRW ASR2000, 2000.

[14] N. Kitaoka and S. Nakagawa, "Evaluation of spectral subtraction with smoothing of time direction on the AURORA2 task," Proc. ICSLP2002, pp.465–468, 2002.

[15] S.F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," IEEE Trans. Acoust. Speech Signal Process., vol.27, no.2, pp.113–120, April 1979.

[16] M. Fujimoto and Y. Ariki, "Combination of temporal domain SVD based speech enhancement and GMM based speech estimation for ASR in noise-evaluation on the AURORA2 task," Proc. Eurospeech 2003, pp.1781–1784, 2003.

[17] C. Uhl and M. Lieb, "Experiments with an extend adaptive SVD enhancement scheme for speech recognition in noise," ICASSP'01, vol.I, pp.280–283, 2001.

[18] J.C. Segura, A. de la Torre, M.C. Benitez, and A.M. Peinado, "Model-based compensation of the additive noise for continuous speech recognition. Experiments using the AURORA II database and tasks," Proc. EUROSPEECH2001, vol.1, pp.221–224, 2001.

[19] M. Ikeda, K. Takeda, and F. Itakura, "Speech enhancement by quadratic comb-filtering," IEICE Technical Report, SP96-45, 1996.

[20] S. Nakamura, K. Takeda, K. Yamamoto, T. Yamada, S. Kuroiwa, N. Kitaoka, T. Nishiura, A. Sasou, M. Mizumachi, C. Miyajima, M. Fujimoto, and T. Endo, "AURORA-2J: An evaluation framework for Japanese noisy speech recognition," IEICE Trans. Inf. & Syst., vol.E88-D, no.3, pp.535–544, March 2005.

[21] T. Yamada, J. Okada, K. Takeda, N. Kitaoka, M. Fujimoto, S. Kuroiwa, K. Yamamoto, T. Nishiura, M. Mizumachi, and S. Nakamura, "Integration of noise reduction algorithms for AURORA2 task," Proc. Eurospeech 2003, pp.1769–1772, 2003.

[22] S. Nakagawa, Pattern Information Processing, Maruzen Ltd., 1999.

Norihide Kitaoka received his B.E. and M.E. degrees from Kyoto University in 1992 and 1994, respectively, and a Dr. Engineering degree from Toyohashi University of Technology in 2000. He joined DENSO CORPORATION, Japan, in 1994. He then joined the Department of Information and Computer Sciences at Toyohashi University of Technology as a Research Associate in 2001 and was a Lecturer from 2003 to 2006. Since 2006 he has been an Associate Professor in the Department of Media Science, Graduate School of Information Science, Nagoya University. His research interests include speech processing, speech recognition, and spoken dialog. He is a member of the Information Processing Society of Japan (IPSJ), the Acoustical Society of Japan (ASJ), and the Japan Society for Artificial Intelligence (JSAI).

Souta Hamaguchi received his B.E. and M.E. degrees from Toyohashi University of Technology in 2004 and 2006, respectively. His research interests include noisy speech recognition. He joined Fuji Xerox in 2006.

Seiichi Nakagawa received Dr. of Eng. degree from Kyoto University in 1977. He joined the Faculty of Kyoto University, in 1976, as a Research Associate in the Department of Information Sciences. He moved to Toyohashi University of Technology in 1980. From 1980 to 1983, he was an Assistant Professor, and from 1983 to 1990 he was an Associate Professor. Since 1990 he has been a Professor in the Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi. From 1985 to 1986, he was a Visiting Scientist in the Department of Computer Science, Carnegie-Mellon University, Pittsburgh, USA. He received the 1997/2001 Paper Award from the IEICE and the 1988 JC Bose Memorial Award from the Institution of Electro. Telecomm. Engrs. His major interests in research include automatic speech recognition/speech processing, natural language processing, human interface, and artificial intelligence. He is a fellow of IPSJ.