Feature Enhancement With Joint Use of Consecutive Corrupted and Noise Feature Vectors With Discriminative Region Weighting

Masayuki Suzuki, *Member, IEEE*, Takuya Yoshioka, *Member, IEEE*, Shinji Watanabe, *Senior Member, IEEE*, Nobuaki Minematsu, *Member, IEEE*, and Keikichi Hirose, *Senior Member, IEEE*

***Abstract*—This paper proposes a feature enhancement method that can achieve high speech recognition performance in a variety of noise environments with feasible computational cost. As the well-known Stereo-based Piecewise Linear Compensation for Environments (SPLICE) algorithm, the proposed method learns piecewise linear transformation to map corrupted feature vectors to the corresponding clean features, which enables efficient operation. To make the feature enhancement process adaptive to changes in noise, the piecewise linear transformation is performed by using a subspace of the joint space of corrupted and noise feature vectors, where the subspace is chosen such that classes (i.e., Gaussian mixture components) of underlying clean feature vectors can be best predicted. In addition, we propose utilizing temporally adjacent frames of corrupted and noise features in order to leverage dynamic characteristics of feature vectors. To prevent overfitting caused by the high dimensionality of the extended feature vectors covering the neighboring frames, we introduce regularized weighted minimum mean square error criterion. The proposed method achieved relative improvements of 34.2% and 22.2% over SPLICE under the clean and multi-style conditions, respectively, on the Aurora 2 task.**

***Index Terms*—Feature enhancement, noise robust automatic speech recognition, non-stationary noise, SPLICE, vector Taylor series.**

I. Introduction

ROBUSTNESS against acoustic noise has been one of the most important issues in research on Automatic Speech Recognition (ASR). When a speech recognizer is employed in noise conditions that differ from those in which the acoustic model of the recognizer was trained, the statistical properties of speech signals obtained in test environments differ significantly from those of the acoustic model of the recognizer. Since this acoustic mismatch greatly degrades ASR performance, tremendous efforts have been made to compensate for this discrepancy over the last few decades [1], [2].

Typical strategies for achieving noise robust ASR include (1) feature normalization, (2) speech enhancement, and (3) model adaptation. The feature normalization approach normalizes certain statistics of feature vectors to eliminate variations caused by acoustic noise. The feature vector typically used by state-of-the-art speech recognizers consists of Mel-Frequency Cepstral Coefficients (MFCCs) and delta cepstra. The normalization processing is applied to both training and test feature vectors. Representative feature normalization techniques include Cepstral Mean Normalization (CMN) [3], RASTA processing [4], and Histogram EQualization (HEQ) [5]. The second approach, speech enhancement, attempts to remove the distortion caused by acoustic noise from the test feature vectors. There are a number of enhancement methods, including Wiener filter [6], SPLICE [7], and Vector Taylor Series (VTS) enhancement [8]–[10]. Finally, the model adaptation methods, such as Parallel Model Combination (PMC) [11] and VTS adaptation [12]–[14], adjust the acoustic model parameters so that the statistical properties of the acoustic model fit well with those of the test feature vectors.

In this paper we focus on a particular class of the second strategy, the feature-based speech enhancement approach (which we will term "feature enhancement"). The goal of this technique is to remove the noise effects from observed corrupted feature vectors. This method can efficiently improve noisy speech recognition performance by leveraging a pre-trained model of clean feature vectors [1], [2], [15]. It is noteworthy that this approach can be combined with feature normalization techniques, yielding even higher recognition performance [5], [6], [16].

Since the relationship between clean feature vectors and the corresponding noise-corrupted vectors is nonlinear, the mapping from the corrupted feature vectors to the corresponding enhanced feature vectors must be described using a nonlinear function. Almost all conventional feature enhancement methods, including SPLICE and VTS enhancement, employ a collection of region-dependent linear transforms to approximate this nonlinearity. Specifically, for a given corrupted feature vector, the corresponding enhanced feature vector is obtained as a weighted average of the feature vectors resulting from the region-specific linear transformations of the corrupted feature vector.

Manuscript received December 28, 2012; revised April 30, 2013; accepted June 12, 2013. Date of publication June 20, 2013; date of current version July 22, 2013. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Dong Yu.

M. Suzuki and N. Minematsu are with the Department of Electrical Engineering and Information Systems, The University of Tokyo, Tokyo 113-8656, Japan (e-mail: suzuki@gavo.t.u-tokyo.ac.jp; suzukimasayuki@gmail.com; mine@gavo.t.u-tokyo.ac.jp).

T. Yoshioka is with the NTT Communication Science Laboratories, Kyoto 619-0237, Japan (e-mail: yoshioka.takuya@lab.ntt.co.jp).

S. Watanabe is with Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA 02139 USA (e-mail: watanabe@merl.com).

K. Hirose is with the Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo, Tokyo 113-8656, Japan (e-mail: hirose@gavo.t.u-tokyo.ac.jp).

Digital Object Identifier 10.1109/TASL.2013.2270407

The key to accurate approximation of the nonlinear function between corrupted and enhanced feature vectors is appropriate partition of the corrupted feature vector space. Considering the fact that the shape of this nonlinear function varies depending on the underlying noise feature vectors, we can say, at the very least, that the feature vector space partition, in other words the way the regions are distributed in the feature vector space, should be modified depending on the noise feature vectors. This implies that methods using a Gaussian Mixture Model (GMM) trained on a set of corrupted feature vectors as in SPLICE are not optimal because the feature vector space partition resulting from such GMMs is fixed and tuned to the training data set. On the other hand, the VTS enhancement method can partition the space appropriately by generating a GMM of corrupted feature vectors using a pre-trained model of clean feature vectors and a noise feature vector model estimated from test utterances. The essential point of this method is that an index of the clean GMM is predicted based on the test corrupted feature vectors. However, a primary problem with VTS enhancement involves computational cost. If we use a time-varying noise model, the computational cost becomes too large, precluding the use of this method in highly non-stationary noise environments.

To overcome the above limitations in conventional feature enhancement methods, this paper proposes a novel concept called Discriminative Region Weighting (DRW). As in the VTS enhancement method, the information from the clean feature vector GMM is utilized when partitioning the feature vector space. The fundamental idea behind avoiding the large computational costs associated with VTS is to use a pre-trained mechanism for predicting a clean GMM component index from an observed feature vector and an estimated noise feature vector. Specifically, instead of directly estimating the clean GMM component index, the method described here first projects a joint feature vector of corrupted speech and noise onto a low-dimensional space, where the separability of the clean GMM components can be maximized. Such a low-dimensional space is found by using Linear Discriminant Analysis (LDA). Then, a GMM in this projected space is used to perform the space partitioning for feature enhancement. In addition, we propose the use of consecutive corrupted and noise feature vectors, which significantly improves performance. We examined the efficacy of our proposed method on Aurora 2 database. Experimental results showed that this method achieved relative improvement of 34.2% and 22.2% over SPLICE under the clean and multi-style conditions, respectively.

The rest of this paper is organized as follows. Section II overviews the algorithms of representative feature enhancement methods. Our proposed method is detailed in Section III. Experimental results are shown in Sections IV and V concludes this paper.

II. Overview of Existing Feature Enhancement Approaches

In this section, we briefly review two representative feature enhancement methods, namely SPLICE [7] and VTS enhancement [9], [10], [13], and point out their limitations. Let us denote frame features of clean speech, noise-corrupted speech, and additive noise as $\boldsymbol{x} = [x_1, \ldots, x_M]^\top$, $\boldsymbol{y} = [y_1, \ldots, y_M]^\top$, and $\boldsymbol{n} = [n_1, \ldots, n_M]^\top$, respectively, where M denotes dimensionality of the feature vectors. We assume that each feature vector consists of MFCCs (including C0) and their velocity and acceleration coefficients. The central task in feature enhancement is to evaluate the posterior probability density function (pdf), $p(\boldsymbol{x}_i|\boldsymbol{y}_i)$, of the clean feature vector at the i-th time-frame $\boldsymbol{x}_i$ given the corresponding corrupted feature vector $\boldsymbol{y}_i$. When we employ the Minimum Mean Squared Error (MMSE) estimation scheme, an estimate of the clean feature vector, $\hat{\boldsymbol{x}}_i$, is calculated as the mean of the posterior probability distribution of $\boldsymbol{x}_i$.

Both SPLICE and VTS enhancement calculate the clean feature posterior pdf $p(\boldsymbol{x}_i|\boldsymbol{y}_i)$ using the "sum-of-products" form as

$$p(\boldsymbol{x}_i|\boldsymbol{y}_i) = \sum_{k=1}^{K} p(k|\boldsymbol{y}_i) p(\boldsymbol{x}_i|\boldsymbol{y}_i, k). \tag{1}$$

Therefore, $\hat{\boldsymbol{x}}_i$ is obtained as

$$\hat{\boldsymbol{x}}_i = \mathbb{E}[\boldsymbol{x}_i|\boldsymbol{y}_i] \tag{2}$$

$$= \sum_{k=1}^{K} p(k|\boldsymbol{y}_i) \mathbb{E}\left[\boldsymbol{x}_i|\boldsymbol{y}_i, k\right], \tag{3}$$

where $\mathbb{E}[a|b]$ denotes an expectation of a given b. This equation means that the space of the corrupted feature vector $\boldsymbol{y}_i$ is divided into K regions, each of which is associated with the region-dependent clean feature estimate given by $\mathbb{E}[\boldsymbol{x}_i|\boldsymbol{y}_i, k]$. The clean feature estimate $\hat{\boldsymbol{x}}_i$ is obtained as the weighted sum of these expectations using $\{p(k|\boldsymbol{y}_i)\}_{k=1\cdots K}$ as weights. SPLICE and VTS enhancement use different strategies for calculating $\{p(k|\boldsymbol{y}_i)\}_{k=1\cdots K}$ and $\{\mathbb{E}[\boldsymbol{x}_i|\boldsymbol{y}_i, k]\}_{k=1\cdots K}$, as described below.

A. SPLICE

SPLICE calculates the region posterior probability $p(k|\boldsymbol{y}_i)$ of (3) by using a pre-trained GMM of corrupted feature vectors, resulting in feature vector space partition tuned to the training environments. Specifically, the corrupted feature vector GMM is trained in advance by using a set of corrupted training data so that we have

$$p(k) = \pi_k^{\boldsymbol{y}} \tag{4}$$

$$p(\boldsymbol{y}_i|k) = \mathcal{N}(\boldsymbol{y}_i; \boldsymbol{\mu}_k^{\boldsymbol{y}}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}}) \tag{5}$$

$$p(\boldsymbol{y}_i) = \sum_{k=1}^{K} p(k) p(\boldsymbol{y}_i|k) = \sum_{k=1}^{K} \pi_k^{\boldsymbol{y}} \mathcal{N}(\boldsymbol{y}_i; \boldsymbol{\mu}_k^{\boldsymbol{y}}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}}), \tag{6}$$

where $\pi_k^{\boldsymbol{y}}, \boldsymbol{\mu}_k^{\boldsymbol{y}}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}}$ are the weight, mean, and diagonal covariance matrix for the k-th component of the GMM, respectively. Using these parameters, we calculate $p(k|\boldsymbol{y}_i)$ as

$$p(k|\boldsymbol{y}_i) = \frac{p(k)p(\boldsymbol{y}_i|k)}{p(\boldsymbol{y}_i)} = \frac{\pi_k^{\boldsymbol{y}} \mathcal{N}(\boldsymbol{y}_i; \boldsymbol{\mu}_k^{\boldsymbol{y}}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}})}{\sum_{k'=1}^{K} \pi_{k'}^{\boldsymbol{y}} \mathcal{N}(\boldsymbol{y}_i; \boldsymbol{\mu}_{k'}^{\boldsymbol{y}}, \boldsymbol{\Sigma}_{k'}^{\boldsymbol{y}})}. \tag{7}$$

On the other hand, SPLICE employs a linear transformation to calculate the region-dependent clean feature vector estimate $\mathbb{E}[\boldsymbol{x}_i|\boldsymbol{y}_i, k]$ of (3). Thus, we obtain the clean feature vector estimate $\hat{\boldsymbol{x}}_i$ according to the following formula

$$\hat{\boldsymbol{x}}_i = \sum_{k=1}^{K} p(k|\boldsymbol{y}_i) \boldsymbol{A}_k \begin{bmatrix} 1 \\ \boldsymbol{y}_i \end{bmatrix}, \tag{8}$$

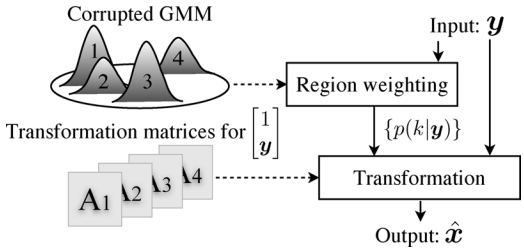

Fig. 1. A diagram of SPLICE.

where each $\boldsymbol{A}_k$ is an $M \times (M+1)$-dimensional transform matrix trained in advance by using pairs of clean and corrupted feature vectors. Fig. 1 illustrates how SPLICE performs feature enhancement.

The set of the transformation matrices $\{\boldsymbol{A}_k\}_{k=1\cdots K}$ is estimated using a separate set of training data in advance by using optimization criteria such as a weighted MMSE criterion. For transformation matrix training, we create a sequence of pairs of clean and corresponding corrupted feature vectors, $\{\boldsymbol{x}_j, \boldsymbol{y}_j\}_{j=1\cdots J}$, where J is the total number of frames in the training data. When using the weighted MMSE criterion, the optimal linear transformation matrix is given by

$$\boldsymbol{A}_k = \underset{\boldsymbol{A}'_k}{\operatorname{argmin}} \sum_{j=1}^{J} p(k|\boldsymbol{y}_j) \left|\left| \boldsymbol{x}_j - \boldsymbol{A}'_k \begin{bmatrix} 1 \\ \boldsymbol{y}_j \end{bmatrix} \right|\right|^2. \tag{9}$$

We can analytically obtain the optimal linear transformation matrix [17] as

$$\boldsymbol{A}_k = \boldsymbol{X}\boldsymbol{P}_k\boldsymbol{Y}'^{\top}(\boldsymbol{Y}'\boldsymbol{P}_k\boldsymbol{Y}'^{\top})^{-1}, \tag{10}$$

where

$$\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_J) \tag{11}$$

$$\boldsymbol{Y}' = \left(\begin{bmatrix} 1 \\ \boldsymbol{y}_1 \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ \boldsymbol{y}_J \end{bmatrix} \right) \tag{12}$$

$$\boldsymbol{P}_k = \begin{pmatrix} p(k|\boldsymbol{y}_1) & 0 & \cdots & 0 \\ 0 & p(k|\boldsymbol{y}_2) & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & p(k|\boldsymbol{y}_J) \end{pmatrix}. \tag{13}$$

The major problem with SPLICE is that it provides insufficient speech recognition performance when training and test noise environments are considerably different or the statistical properties of noise feature vectors may change significantly with time. The reason for this drawback is as follows. Considering the fact that the shape of the nonlinear function between the clean and corrupted feature vectors varies depending on the intervening noise feature vectors, the nonlinear function should be changed depending on the noise feature vectors. SPLICE does not utilize the noise features, however, but rather only the GMM of corrupted features and the transformation matrices that are tuned to the training noise environment.

B. VTS Enhancement

Unlike SPLICE, VTS enhancement employs a noise-adaptive GMM of corrupted feature vectors to partition the feature vector space depending on the statistical properties of latent noise feature vectors. To create such a noise-adaptive GMM, this approach combines a pre-trained clean feature vector GMM and a noise feature vector model obtained from test utterances. The clean GMM specifies $p(\boldsymbol{x}_i)$ as follows

$$p(k) = \pi_k^{\boldsymbol{x}} \tag{14}$$

$$p(\boldsymbol{x}_i|k) = \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}) \tag{15}$$

$$p(\boldsymbol{x}_i) = \sum_{k=1}^{K} p(k)p(\boldsymbol{x}_i|k) = \sum_{k=1}^{K} \pi_k^{\boldsymbol{x}} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}}). \tag{16}$$

On the other hand the distribution of the noise feature vectors is often modeled using a single Gaussian. Thus, we have

$$p(\boldsymbol{n}_i) = \mathcal{N}(\boldsymbol{n}_i; \boldsymbol{\mu}^{\boldsymbol{n}_i}, \boldsymbol{\Sigma}^{\boldsymbol{n}_i}). \tag{17}$$

Here, the simplest approach to estimating $\boldsymbol{\mu}^{\boldsymbol{n}_i}$ and $\boldsymbol{\Sigma}^{\boldsymbol{n}_i}$ would be to calculate the sample average and variance of the feature vectors obtained from non-speech frames. It is also possible to update these parameters by jointly using speech and non-speech frames [13], [18], [19].

To create the corrupted feature vector model from the above clean and noise models, we also have to explicitly describe the relationship among corrupted speech $\boldsymbol{y}_i$, clean speech $\boldsymbol{x}_i$, and noise $\boldsymbol{n}_i$. Most conventional methods use the following model

$$\boldsymbol{y_i} = \boldsymbol{x}_i + \boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{n}_i) \tag{18}$$

$$\boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{n}_i) = \boldsymbol{D}\log\left(\boldsymbol{1} + \exp(\boldsymbol{C}\boldsymbol{n}_i - \boldsymbol{C}\boldsymbol{x}_i)\right), \tag{19}$$

where $\boldsymbol{g}$ is called a mismatch function, $\boldsymbol{D}$ is a discrete cosine transformation conversion matrix, $\boldsymbol{C}$ is its inverse, $\log$ and $\exp$ are vector functions that calculate the logarithm and exponent, respectively, of each vector element. The feature vectors, $\boldsymbol{x}_i$, $\boldsymbol{y}_i$, and $\boldsymbol{n}_i$, are assumed to be represented in the static MFCC domain.

Using the clean speech GMM and the noise feature vector model, VTS enhancement creates a GMM of corrupted feature vectors and exploits it to perform feature enhancement as described below. To avoid the difficulty resulting from directly manipulating the nonlinear function $\boldsymbol{g}$ in statistical inference, we use the first order VTS approximation to linearize the nonlinear mismatch function $\boldsymbol{g}$ with respect to $\boldsymbol{x}_i$ and $\boldsymbol{n}_i$ as

$$\begin{aligned}\boldsymbol{g}(\boldsymbol{x}_i, \boldsymbol{n}_i; \boldsymbol{x}^0, \boldsymbol{n}^0) \approx\ & \boldsymbol{A}[\boldsymbol{x}^0, \boldsymbol{n}^0](\boldsymbol{x}_i - \boldsymbol{x}^0) \\ & + \boldsymbol{B}[\boldsymbol{x}^0, \boldsymbol{n}^0](\boldsymbol{n}_i - \boldsymbol{n}^0) + \boldsymbol{g}(\boldsymbol{x}^0, \boldsymbol{n}^0),\end{aligned} \tag{20}$$

where $\boldsymbol{x}^0$ and $\boldsymbol{n}^0$ are the linearization points, and $\boldsymbol{A}[\boldsymbol{x}^0, \boldsymbol{n}^0]$ and $\boldsymbol{B}[\boldsymbol{x}^0, \boldsymbol{n}^0]$ are the first partial derivatives of the mismatch function $\boldsymbol{g}(\boldsymbol{x}^0, \boldsymbol{n}^0)$ with regard to $\boldsymbol{x}_i$ and $\boldsymbol{n}_i$, respectively. It should be noted that both $\boldsymbol{A}$ and $\boldsymbol{B}$ are full matrices. By using this approximation, $p(\boldsymbol{y}_i|k)$ of the corrupted feature vector GMM is obtained as follows

$$p(\boldsymbol{y}_i|k) = \mathcal{N}(\boldsymbol{y}_i; \boldsymbol{\mu}_k^{\boldsymbol{y}_i}, \boldsymbol{\Sigma}_k^{\boldsymbol{y}_i}) \tag{21}$$

$$\boldsymbol{\mu}_k^{\boldsymbol{y}_i} \approx \boldsymbol{\mu}_k^{\boldsymbol{x}} + \boldsymbol{g}(\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\mu}^{\boldsymbol{n}_i}) \tag{22}$$

$$\begin{aligned}\boldsymbol{\Sigma}_k^{\boldsymbol{y}_i} \approx\ & \boldsymbol{A}[\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\mu}^{\boldsymbol{n}_i}]\boldsymbol{\Sigma}_k^{\boldsymbol{x}}\boldsymbol{A}[\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\mu}^{\boldsymbol{n}_i}]^{\top} \\ & + \boldsymbol{B}[\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\mu}^{\boldsymbol{n}_i}]\boldsymbol{\Sigma}^{\boldsymbol{n}_i}\boldsymbol{B}[\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\mu}^{\boldsymbol{n}_i}]^{\top}.\end{aligned} \tag{23}$$

Note that in (21)–(23) we have assumed that the linearization point of the first-order VTS approximation consists of the clean speech mean $\boldsymbol{\mu}_k^{\boldsymbol{x}}$ and the noise mean $\boldsymbol{\mu}^{\boldsymbol{n}_i}$. In addition, $p(k)$ can

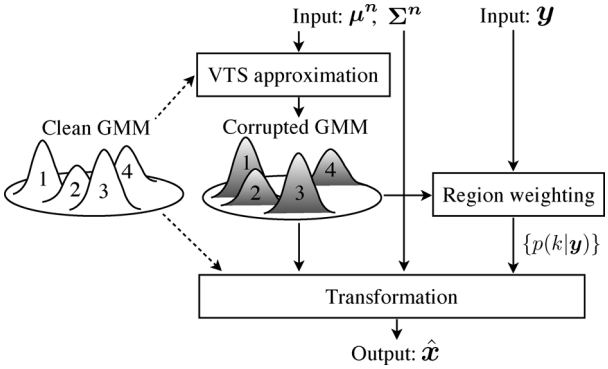

Fig. 2. A diagram of VTS enhancement.

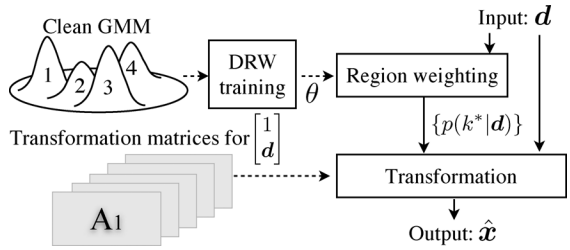

Fig. 3. A diagram of our proposed method.

be approximated by the prior weight of the clean feature vector GMM so that

$$p(k) = \pi_k^{\boldsymbol{y}_i} \approx \pi_k^{\boldsymbol{x}}. \tag{24}$$

Using the above parameters and an observed corrupted feature vector $\boldsymbol{y}_i$, VTS enhancement calculates the MMSE estimate $\hat{\boldsymbol{x}}_i$ of the clean feature vector

$$\hat{\boldsymbol{x}}_i = \sum_{k=1}^{K} p(k|\boldsymbol{y}_i) \underbrace{\left(\boldsymbol{\mu}_k^{\boldsymbol{x}} + \boldsymbol{\Sigma}_k^{\boldsymbol{x}} \boldsymbol{A}[\boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\mu}^{\boldsymbol{n}_i}]^\top {\boldsymbol{\Sigma}_k^{\boldsymbol{y}_i}}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}_k^{\boldsymbol{y}_i})\right)}_{\text{corresponding to } \mathbb{E}[\boldsymbol{x}_i|\boldsymbol{y}_i,k] \text{ in (3)}}. \tag{25}$$

Fig. 2 illustrates how the VTS enhancement approach performs feature enhancement.

Previous work has shown that VTS enhancement can retain a high degree of recognition accuracy regardless of noise environments [9], [10]. This results from the mechanism of VTS enhancement utilizing the noise-adaptive corrupted feature vector GMM, which is generated by combining the clean GMM and the noise feature vector model. The use of noise-adaptive GMM enables us to deal appropriately with the nonlinear relationship between the clean and corrupted feature vectors. Therefore, VTS enhancement can prevent recognition performance from degrading in unseen test noise environments.

However, one drawback to VTS enhancement is its computational cost. We have to calculate K inverse matrices $\{{\boldsymbol{\Sigma}_k^{\boldsymbol{y}_i}}^{-1}\}_{k=1\cdots K}$ to calculate $\{p(k|\boldsymbol{y}_i)\}_{k=1\cdots K}$ every time the noise model is updated. $\boldsymbol{\Sigma}_k^{\boldsymbol{y}_i}$ is a full covariance matrix and calculating its inverse matrix requires tremendous computational cost when the noise model is updated every single frame. A time-invariant noise model is therefore necessary, which could result in insufficient speech recognition performance in nonstationary noise environments.

III. Proposed Method

To efficiently follow changes in noise feature vector statistics, the proposed method combines the following two concepts: pre-training of mapping and prediction of clean GMM index. To this end, we assume that each corrupted feature vector $\boldsymbol{y}_i$ is associated with the clean GMM index k_i^* calculated as

$$k_i^* = \operatorname*{argmax}_k p(k|\boldsymbol{x}_i) \tag{26}$$

$$= \operatorname*{argmax}_k \frac{\pi_k^{\boldsymbol{x}} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}})}{\sum_{k=1}^{K} \pi_k^{\boldsymbol{x}} \mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\mu}_k^{\boldsymbol{x}}, \boldsymbol{\Sigma}_k^{\boldsymbol{x}})}, \tag{27}$$

where $\boldsymbol{x}_i$ is the clean feature vector corresponding to $\boldsymbol{y}_i$.

The central idea behind the proposed method is to calculate the posterior probabilities of the underlying clean GMM indices as in VTS enhancement by using a posterior model $p(k = k_i^*|\boldsymbol{y}_i, \boldsymbol{\mu}^{\boldsymbol{n}_i})$ that is directly trained on a set of training data. This allows efficient prediction of the clean GMM index from corrupted and noise feature vectors. We use this model as an alternative to $p(k|\boldsymbol{y}_i)$ in (3). This idea is quite natural when we recall that the aim of feature enhancement is to obtain estimates of clean features.

In order to improve the posterior estimation accuracy by using dynamic speech characteristics, we further propose using the corrupted and noise feature vectors covering several adjacent frames. Thus, when we let $2M(2N^R + 1)$-dimensional vector $\boldsymbol{d}_i^{N^R}$ denotes

$$\boldsymbol{d}_i^{N^R} = \left[\begin{bmatrix} \boldsymbol{y}_{i-N^R} \\ \boldsymbol{\mu}^{\boldsymbol{n}_{i-N^R}} \end{bmatrix}^\top, \ldots, \begin{bmatrix} \boldsymbol{y}_i \\ \boldsymbol{\mu}^{\boldsymbol{n}_i} \end{bmatrix}^\top, \ldots, \begin{bmatrix} \boldsymbol{y}_{i+N^R} \\ \boldsymbol{\mu}^{\boldsymbol{n}_{i+N^R}} \end{bmatrix}^\top \right]^\top, \tag{28}$$

we predict the clean GMM index with the posterior model $p(k = k_i^*|\boldsymbol{d}_i^{N^R})$. We call $\boldsymbol{d}_i^{N^R}$ an extended feature vector.

Specifically, we propose calculating the posterior probability of the clean GMM index $p(k = k_i^*|\boldsymbol{d}_i^{N^R})$ by using our proposed method of Discriminative Region Weighting (DRW)

$$p(k = k_i^*|\boldsymbol{d}_i^{N^R}) = f(k = k_i^*|\boldsymbol{d}_i^{N^R}; \theta), \tag{29}$$

where f is a discriminatively trained model for estimating k_i^* based on $\boldsymbol{d}_i^{N^R}$, and θ is a parameter set of the model learned from training data.

Finally, we estimate a collection of mappings from the extended feature vectors to the corresponding clean feature vectors. Inspired by SPLICE, we propose using linear transformation matrices as

$$\hat{\boldsymbol{x}}_i = \sum_{k=1}^{K} f(k = k_i^*|\boldsymbol{d}_i^{N^R}; \theta) \boldsymbol{A}_k \begin{bmatrix} 1 \\ \boldsymbol{d}_i^{N^T} \end{bmatrix}, \tag{30}$$

where $\{\boldsymbol{A}_{k^*}\}_{k^*=1\cdots K}$ are trained in advance by using a weighted MMSE criterion as in SPLICE. Note that the frame number N^T for transformation can be different from that for region weighting given by N^R when computing $\boldsymbol{d}_i^{N^T}$ in the way shown in (28). Fig. 3 illustrates how feature enhancement is performed by our proposed method.

TABLE I
A COMPARISON OF SPLICE, VTS ENHANCEMENT, AND THE PROPOSED METHOD.

	Region-weighting step	Transformation step
General form (3)	$p(k\|\boldsymbol{y})$	$\mathbb{E}[\boldsymbol{x}\|\boldsymbol{y},k]$
SPLICE (8)	GMM of $\boldsymbol{y}$	Linear trans. of $\boldsymbol{y}$
VTS enhancement (25)	VTS from GMM of $\boldsymbol{x}$	MMSE estimation
Proposed method (30)	DRW from GMM of $\boldsymbol{x}$	Linear trans. of $\boldsymbol{d}$

It would be interesting to compare DRW with fMPE [20], MMI-SPLICE [21], and MPE-HLDA [22], all of which leverage the "sum-of-products" form and utilize discriminative approaches. A key difference between DRW and these known techniques is that the former aims at predicting clean GMM indices from corrupted feature vectors, whereas the latter optimize the parameters involved to directly improve speech recognition performance. As for the use of consecutive feature vectors, similar ideas have already been adopted in fMPE and MPE-HLDA, both of which use posteriors calculated over several adjacent frames [20], [22].

In summary, our proposed method differs from SPLICE in two ways. First, in the region-weighting step, the proposed method estimates the posterior probability of the clean feature GMM index k_i^*, while SPLICE utilizes a corrupted feature vector GMM. Secondly, in the transformation step, the proposed method uses corrupted and noise feature vectors obtained from the current frame and several adjacent frames, while SPLICE uses only a corrupted feature vector at the current frame. Table I shows a summary of the differences between SPLICE, VTS enhancement, and the proposed method. Below, we present one representative implementation of the DRW concept in detail.

A. Region-Weighting Step

The objective of the region-weighting step is to calculate the posterior probability that each clean GMM index k has produced the underlying clean feature vector $\boldsymbol{x}_i$, given an extended feature vector $\boldsymbol{d}_i^{N^R}$. We consider this to be a classification task. To train the posterior model $p(k = k_i^*|\boldsymbol{d}_i^{N^R})$, we create a set of training data, consisting of tuples of an extended feature vector $\boldsymbol{d}_i^N$, the corresponding clean feature vector $\boldsymbol{x}_i$, and the corresponding clean GMM weight vector $q_i = [q_{1,i}, \ldots, q_{K,i}]$. The clean GMM weight vector for each training sample i is calculated by first training a clean feature vector GMM and then calculating $q_{k,i} = p(k|\boldsymbol{x}_i)$. The posterior probability $p(k = k_i^*|\boldsymbol{d}_i^{N^R}; \theta)$ can be modeled using a variety of discriminative models such as the Support Vector Machine (SVM) and log linear model.

As an alternative to such conventional discriminative models, we propose applying an LDA-based dimensionality reduction to $\boldsymbol{d}_i^{N^R}$ by using the clean GMM indices as the labels for LDA instead of directly estimating $p(k = k_i^*|\boldsymbol{d}_i^{N^R}; \theta)$ with those conventional discriminative models. After finding the LDA transformation of $\boldsymbol{d}_i^{N^R}$, we construct another GMM of the compressed feature vectors and use the posterior probabilities of the indices of this GMM to carry out the region-weighting step. Although the indices are generally different from the indices of the clean GMM, the GMM indices calculated as above will be able to provide an appropriate feature space partition because the separability of the clean GMM components have been maximized in the compressed feature space. Our preliminary experiments on the Aurora 2 database showed that DRW using this LDA-GMM approach was comparable to the SVM-based DRW with a linear kernel using optimized hyperparameters. Since the LDA-GMM approach is more computationally efficient in the training phase, we report the results obtained with the LDA-GMM-based DRW.

Here we explain the detail of the LDA-GMM method. First, we estimate the LDA-based dimensionality reduction matrix $\boldsymbol{L}$ to find a low-dimensional subspace of the extended feature vectors suitable to discriminate k^*. The dimensionality reduction matrix $\boldsymbol{L}$ is calculated by using training data set $\{\{q_{k,i}\}_{k=1\cdots K}, \boldsymbol{d}_j\}_{i=1\cdots J}$ as

$$\boldsymbol{L} = \underset{\boldsymbol{W}}{\operatorname{argmin}} \frac{\boldsymbol{W}^\top \boldsymbol{\Sigma}^w \boldsymbol{W}}{\boldsymbol{W}^\top \boldsymbol{\Sigma}^b \boldsymbol{W}}, \tag{31}$$

where

$$\boldsymbol{\Sigma}^w = \sum_{k=1}^{K} \sum_{j=1}^{J} q_{k,j} \left(\boldsymbol{d}_j^{N^R} - \boldsymbol{\mu}_k^w\right)\left(\boldsymbol{d}_j^{N^R} - \boldsymbol{\mu}_k^w\right)^\top \tag{32}$$

$$\boldsymbol{\Sigma}^b = \sum_{k=1}^{K} \left(\sum_{j=1}^{J} q_{k,j}\right) \left(\boldsymbol{\mu}_k^w - \frac{\sum_j \boldsymbol{d}_j^{N^R}}{J}\right)\left(\boldsymbol{\mu}_k^w - \frac{\sum_j \boldsymbol{d}_j^{N^R}}{J}\right)^\top \tag{33}$$

$$\boldsymbol{\mu}_k^w = \frac{1}{\sum_{j=1}^{J} q_{k,j}} \sum_{j=1}^{J} q_{k,j} \boldsymbol{d}_j^{N^R}. \tag{34}$$

The analytical solution for (31) is obtained as the eigenvectors of $(\boldsymbol{\Sigma}^w)^{-1}\boldsymbol{\Sigma}^b$ with the P smallest eigenvectors.

Then, we train an S-component GMM of the compressed feature vectors given by $\boldsymbol{v}_i = \boldsymbol{L}\boldsymbol{d}_i^{N^R}$ as

$$p(s) = \pi_s^{\boldsymbol{v}} \tag{35}$$

$$p(\boldsymbol{v}_i|s) = \mathcal{N}(\boldsymbol{v}_i; \boldsymbol{\mu}_s^{\boldsymbol{v}}, \boldsymbol{\Sigma}_s^{\boldsymbol{v}}) \tag{36}$$

$$p(\boldsymbol{v}_i) = \sum_{s=1}^{S} \pi_s^{\boldsymbol{v}} \mathcal{N}(\boldsymbol{v}_i; \boldsymbol{\mu}_s^{\boldsymbol{v}}, \boldsymbol{\Sigma}_s^{\boldsymbol{v}}), \tag{37}$$

where $\pi_s, \boldsymbol{\mu}_s^{\boldsymbol{v}}$, and $\boldsymbol{\Sigma}_s^{\boldsymbol{v}}$, are the weight, mean, and diagonal covariance matrix, respectively, of the s-th Gaussian. Consequently, we calculate

$$p(s|\boldsymbol{d}_i^{N^R}) = \frac{\pi_s^{\boldsymbol{v}} \mathcal{N}(\boldsymbol{v}_i; \boldsymbol{\mu}_s^{\boldsymbol{v}}, \boldsymbol{\Sigma}_s^{\boldsymbol{v}})}{\sum_{s=1}^{S} \pi_s^{\boldsymbol{v}} \mathcal{N}(\boldsymbol{v}_i; \boldsymbol{\mu}_s^{\boldsymbol{v}}, \boldsymbol{\Sigma}_s^{\boldsymbol{v}})}, \tag{38}$$

and utilize the $\{p(s|\boldsymbol{d}_i^{N^R})\}_{s=1\cdots S}$ for the region-weighting step. Thus, (38) is used in place of (29) although s is in general different from k.

B. Transformation Step

In the transformation step, we calculate an enhanced feature vector $\hat{\boldsymbol{x}}$ as a weighted average of region-dependent linear transforms of $\boldsymbol{d}_i^{N^T}$ as

$$\hat{\boldsymbol{x}}_i = \sum_{s=1}^{S} p(s|\boldsymbol{d}_i^{N^R}) \boldsymbol{A}_s \begin{bmatrix} 1 \\ \boldsymbol{d}_i^{N^T} \end{bmatrix}. \tag{39}$$

Each $M \times (2M(2N^T+1)+1)$-dimensional transform matrix $\boldsymbol{A}_s$ is estimated in advance by using a set of training data consisting of pairs of an extended joint feature vector $\boldsymbol{d}_i^{N^T}$ and the corresponding clean feature vector $\boldsymbol{x}_i$. Specifically, $\boldsymbol{A}_s$ is estimated by using the following regularized weighting MMSE estimation scheme

$$\boldsymbol{A}_s = \underset{\boldsymbol{A}_s}{\operatorname{argmin}} \sum_{j=1}^{J} p(s|\boldsymbol{d}_i^{N^R}) \left\| \boldsymbol{x}_j - \boldsymbol{A}_s \begin{bmatrix} 1 \\ \boldsymbol{d}_j^{N^T} \end{bmatrix} \right\|^2 - \lambda R_s \quad (40)$$

$$R_s = \frac{\mathbf{1}^\top \boldsymbol{A}_{s'}^\top \operatorname{diag}(\boldsymbol{D}'\boldsymbol{P}_s\boldsymbol{D}'^\top)\boldsymbol{A}'_s\mathbf{1}}{M} \quad (41)$$

$$\boldsymbol{D}' = \left(\begin{bmatrix} 1 \\ \boldsymbol{d}_1^{N^T} \end{bmatrix}, \ldots, \begin{bmatrix} 1 \\ \boldsymbol{d}_J^{N^T} \end{bmatrix} \right) \quad (42)$$

$$\boldsymbol{P}_s = \begin{pmatrix} p(s|\boldsymbol{d}_1^{N^R}) & 0 & \cdots & 0 \\ 0 & p(s|\boldsymbol{d}_2^{N^R}) & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & p(s|\boldsymbol{d}_J^{N^R}) \end{pmatrix}, \quad (43)$$

where the second term on the right-hand side of (40) is an L2 regularization term for preventing the portion of $\boldsymbol{A}_s$ corresponding to $\boldsymbol{d}_i^{N^T}$ from over-fitting the training data caused by the high dimensionality of the feature vector space. Considering that the bias component of $\boldsymbol{A}_s$ corresponding to 1 has dominant effect for feature enhancement [7], we do not regularize this component. λ is a pre-determined regularization parameter, $\mathbf{1}$ is an M-dimensional column vector, and $\boldsymbol{A}'_s$ is the lower $M \times 2M(2N^T+1)$ portion of $\boldsymbol{A}_s$, corresponding to $\boldsymbol{d}_i^{N^T}$. The term $\operatorname{diag}(\boldsymbol{D}'\boldsymbol{P}_s\boldsymbol{D}'^\top)$ in (41) is the regularization term used to exert an equal effect on all the dimensions, where $\operatorname{diag}(\cdot)$ leaves diagonal elements intact but changes the others to 0. The analytical solution for (40) can be obtained by

$$\boldsymbol{A}_s = \boldsymbol{X}\boldsymbol{P}_s\boldsymbol{D}'^\top(\boldsymbol{D}'\boldsymbol{P}_s\boldsymbol{D}'^\top + \lambda \boldsymbol{I}' \operatorname{diag}(\boldsymbol{D}'\boldsymbol{P}_s\boldsymbol{D}'^\top))^{-1}, \quad (44)$$

where $\boldsymbol{I}'$ the $(2M(2N^T+1)+1) \times (2M(2N^T+1)+1)$ matrix with ones at the second-to-last diagonal elements and zeros elsewhere.

C. Comparison With NMN-SPLICE

Noise Mean Normalized (NMN)-SPLICE was proposed as a heuristic solution to mitigate the SPLICE drawback, namely that the noise compensation effect of SPLICE is degraded when training and test environments are different [7], [23]. NMN-SPLICE uses the following formula to obtain a clean feature estimate $\hat{\boldsymbol{x}}$

$$\hat{\boldsymbol{x}} = \sum_{k=1}^{K} p(k|\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}) \boldsymbol{A}_k \begin{bmatrix} 1 \\ \boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}} \end{bmatrix} + \boldsymbol{\mu}^{\boldsymbol{n}}, \quad (45)$$

where $\{p(k|\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}})\}_{k=1\cdots K}$ are calculated by using a pre-trained GMM of the vectors given by $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$, and $\{\boldsymbol{A}_k\}_{k=1\cdots K}$ are estimated using the weighted MMSE criterion as in SPLICE. The important difference between the original SPLICE technique and NMN-SPLICE is that the latter uses $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$ instead of $\boldsymbol{y}$ for region-weighting and linear transformation. The use of $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$ was shown to remove the effect of noise variability from the variability of corrupted features to some extent, thereby helping us accurately capture the relationship between the corrupted and clean feature vectors, leading to higher speech recognition accuracy.

NMN-SPLICE can be interpreted as a special case of the proposed method. As for the region-weighting step, if we set $N^R = 0$ and $\boldsymbol{L} = [\boldsymbol{I}, -\boldsymbol{I}]$, $\boldsymbol{v} = \boldsymbol{L}\boldsymbol{d}$ becomes $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$ and therefore the posterior computation of the proposed method reduces to that of NMN-SPLICE. The transformation step of the proposed method is also a generalization of that of NMN-SPLICE. To see this, let us set $N^T = 0$ and denote the sub-matrices of $\boldsymbol{A}_k$ corresponding to $\boldsymbol{y}$ and $\boldsymbol{\mu}^{\boldsymbol{n}}$ by $\boldsymbol{A}_k^{\boldsymbol{y}}$ and $\boldsymbol{A}_k^{\boldsymbol{\mu}_{\boldsymbol{n}}}$, respectively. Then, we can see that forcing $\boldsymbol{A}_k^{\boldsymbol{\mu}_{\boldsymbol{n}}}$ to be equal to $-\boldsymbol{A}_k^{\boldsymbol{y}}$ makes the proposed transformation step equivalent to that of NMN-SPLICE.

IV. Experiment

A. Experimental Setup

The proposed feature enhancement method was evaluated on the Aurora 2 database [24]. The Aurora 2 task involves recognizing connected English digits corrupted by additive and convolutional noise. The task consists of three types of test sets. Test set A contains 4 sets of 1001 utterances, with each set corrupted by either subway, babble, airport, or exhibition hall noise signals, at signal-to-noise ratios (SNRs) of ∞, 20, 15, 10, 5, 0, and -5 dBs. Test set B also contains 4 sets, each corrupted by either restaurant, street, airport, and train station noise signals. Test set C contains 2 sets of 1001 utterances, corrupted by the convolutional noise produced by an MIRS filter [24] and subway and street noise signals at the same range of SNRs as the test sets A and B.

The Aurora 2 task involves two different acoustic model training conditions: clean and multi-style conditions. In the clean condition, the acoustic model is trained on clean data, while in the multi-style condition, the acoustic model is trained on data with various SNRs. The training data for the multi-style condition consist of speech signals corrupted by subway, babble, airport, and exhibition hall noise signals at SNRs of ∞, 20, 15, 10, and 5 dBs. Therefore, the noise environments of test set A are identical to those of the training data set, while test sets B and C use noise environments that differ from the training environments. The acoustic models were trained using the complex back-end recipe of Aurora 2 [25]. A hidden Markov model with 16 states per digit and 20 Gaussian mixtures per state was created for each digit as a whole word.

In order to estimate the linear transformation matrices $\{\boldsymbol{A}_k\}_{k=1,\ldots,K}$, the LDA matrix $\boldsymbol{L}$, and the GMM parameters $\{\pi_s, \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s\}_{s=1,\ldots,S}$ used in performing SPLICE, NMN-SPLICE, and the proposed method, we employed the multi-style training data set regardless of whether the acoustic model was trained on the clean or multi-style data.

As a feature vector for speech recognition, we used a 39-dimensional vector consisting of 13 MFCCs (including C0) and their velocity and acceleration parameters. This means that the dimension of the joint feature vector $[\boldsymbol{y}^\top \boldsymbol{\mu}_{\boldsymbol{n}}^\top]^\top$ is 78. In order to compute the extended feature vector $\boldsymbol{d}$, we used the left and right N frames, resulting in a $(2N+1)*78$-dimensional extended feature vector. In the experiment, we set N to 0 and 4. We set the dimension of the subspace found by LDA P to 39 to compare

TABLE II
AVERAGE WORD ERROR RATES (%) OBTAINED BY VARIOUS FEATURE ENHANCEMENT METHODS FOR AURORA 2.

#	Region-weighting step	Transformation step	Note	Clean condition				Multi condition			
				Set A	Set B	Set C	Ave.	Set A	Set B	Set C	Ave.
1	-	-	No enhancement	30.62	25.67	30.05	28.52	7.80	7.95	7.53	7.80
2	GMM of $\boldsymbol{y}$	$\boldsymbol{A}_k[1\ \boldsymbol{y}^\top]^\top$	SPLICE	10.73	12.51	14.06	12.11	6.73	11.15	9.10	8.97
3	GMM of $\boldsymbol{y}$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{0\top}]^\top(\lambda=0)$		9.31	10.78	12.29	10.49	5.85	10.24	8.92	8.22
4	GMM of $\boldsymbol{y}$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{4\top}]^\top(\lambda=10^{-3})$		8.78	9.81	11.45	9.73	**5.57**	8.91	8.74	7.54
5	GMM of $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$	$\boldsymbol{A}_k[1\ (\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}})^\top]^\top+\boldsymbol{\mu}^n$	NMN-SPLICE	10.07	10.18	10.46	10.39	6.61	8.29	6.56	7.27
6	GMM of $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$	$\boldsymbol{A}_k[1\ \boldsymbol{y}^\top]^\top$		10.30	10.32	11.69	10.59	7.00	8.66	7.54	7.77
7	GMM of $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{0\top}]^\top(\lambda=0)$		9.36	9.46	10.40	9.61	6.46	8.13	6.56	7.15
8	GMM of $\boldsymbol{y}-\boldsymbol{\mu}^{\boldsymbol{n}}$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{4\top}]^\top(\lambda=10^{-3})$		7.98	8.47	**9.22**	8.42	6.07	7.78	**6.52**	6.84
9	DRW by LDA-GMM of $\boldsymbol{d}^0$	$\boldsymbol{A}_k[1\ \boldsymbol{y}^\top]^\top$		8.99	9.45	12.92	9.96	6.88	8.84	9.67	8.22
10	DRW by LDA-GMM of $\boldsymbol{d}^0$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{0\top}]^\top(\lambda=0)$		8.21	8.51	11.50	8.99	6.54	8.20	8.62	7.62
11	DRW by LDA-GMM of $\boldsymbol{d}^0$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{4\top}]^\top(\lambda=10^{-3})$	Best for multi	7.25	7.52	10.11	7.93	5.61	**7.31**	7.58	**6.68**
12	DRW by LDA-GMM of $\boldsymbol{d}^4$	$\boldsymbol{A}_k[1\ \boldsymbol{y}^\top]^\top$		7.81	8.20	10.81	8.57	7.21	8.71	9.75	8.32
13	DRW by LDA-GMM of $\boldsymbol{d}^4$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{0\top}]^\top(\lambda=0)$		7.20	7.56	10.09	7.92	6.47	7.98	8.88	7.56
14	DRW by LDA-GMM of $\boldsymbol{d}^4$	$\boldsymbol{A}_k[1\ \boldsymbol{d}^{4\top}]^\top(\lambda=10^{-3})$	Best for clean	**6.50**	**7.19**	9.42	**7.36**	6.06	7.37	8.17	7.01

the proposed method with SPLICE and NMN-SPLICE in a fair manner. The mixture numbers (K and S) of the GMMs were both set to 1024 because the performance of SPLICE was not significantly improved with larger GMMs. We did not perform environment selection for either (NMN-)SPLICE or the proposed method because significant improvement could not be obtained, particularly for set B. Consequently, the numbers of free parameters used in the region-weighting step are 80,896, 82,417 and 204,097 for SPLICE, DRW with $N^R = 0$, and DRW with $N^R = 4$, respectively. Those used in the transformation step were 1,597,440, 3,154,944, and 28,075,008 for SPLICE, DRW with $N^T = 0$, and DRW with $N^T = 4$, respectively. The regularization parameter λ was set at 0 and 10^{-3} for $N^T = 0$ and 4, respectively. After the enhancement process, we applied cepstrum mean normalization on an utterance-by-utterance basis.

To obtain estimates of noise feature vectors, we employed the method proposed in [19], which was shown to be effective even in highly non-stationary noise environments. Since this method estimates log mel frequency spectra of noise signals, we converted the estimated log mel frequency features to the corresponding static and dynamic MFCCs.

B. Experimental Results

Table II shows the averages of Word Error Rates (WERs) for each feature enhancement method. The WERs were averaged over all noise types and SNRs between 0 and 20 dBs. Conditions #1, #2, and #5 show the WERs that were obtained without feature enhancement, with SPLICE, and with NMN-SPLICE, respectively. The others (#3, #4, and #6 to #14) show the WERs that were obtained with different configurations of the region-weighting and transformation steps, including our proposed method.

First, we discuss the effectiveness of the transformation step. Compared with the methods using only the corrupted feature vectors for the transformation step, i.e., #2, #6, #9, and #12 in Table II, the methods using both the corrupted and estimated noise feature vectors, i.e., #3, #7, #10, and #13, achieved lower WERs in all sets and conditions. In addition, the setup indicated by #7 outperformed NMN-SPLICE. These results clearly show the effectiveness of jointly using the corrupted and noise feature vectors in the transformation step. Comparing the methods using only the current frame for estimating the clean feature vector at the same frame, i.e., #3, #7, #10, and #13, with those using the current and several adjacent frames, i.e., #4, #8, #11, and #14, the latter always outperformed the former. This result shows the effectiveness of the use of the feature vector context obtained from those adjacent frames.

Next, we discuss the effectiveness of the region-weighting step. Compared with the methods using the GMMs of corrupted feature vectors, i.e., #2 to #4, the methods using the GMMs of the differences between corrupted and noise feature vectors, i.e., #6 to #8, achieved lower WERs. The effectiveness of the latter approach was pronounced in test set B. This result clearly shows that noise mean normalization for the region-weighting step effectively mitigated the sensitivity of SPLICE to the variability of noise environments, which is consistent with [7]. When we compared the methods that use the GMMs of the differences between corrupted and noise feature vectors, i.e., #6 to #8, with the methods using the proposed DRW, i.e., #9 to #14, the latter always outperformed the former in the clean condition, while the performance of the DRW approach was almost equal to or even slightly surpassed by that of the GMM approach in the multi-style condition. Comparing the methods using only the current frame for DRW, i.e., #9 to #11, with those using the current and several adjacent frames, i.e., #12 to #14, the latter outperformed the former in the clean condition, whereas the use of the acoustic context decreased the speech recognition performance in the case of multi-style training. This means that the use of the acoustic context from neighboring frames certainly contributed to reducing the mismatch between the clean acoustic model and the corrupted feature vectors. However, in general, feature vectors that are closer to the clean acoustic model on average do not necessarily fit better with GMM-HMM models. This could be one reason that use of the acoustic context did not improve the speech recognition performance in the multi-style condition. Since the optimization criterion for DRW is different from the likelihood of the acoustic model, the configuration that performs the best in the clean condition may not be optimal for the multi-style condition. On the other hand, it should be noted that using neighboring frames in the transformation step always resulted in smaller WERs. We should note that result #14 in the clean condition of set B was superior to all other results

TABLE III
SUMMARY OF ALL RESULTS BY NMN-SPLICE (#5 IN TABLE II) FOR AURORA 2.

Clean condition													
	A					B					C		
SN	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway	Street	Average
∞	1.29	1.15	1.43	1.05	1.23	1.29	1.15	1.43	1.05	1.23	1.32	1.03	1.17
20	0.92	0.70	0.60	0.93	0.79	0.80	1.03	0.78	0.56	0.79	1.01	1.24	1.12
15	1.66	1.42	1.10	1.82	1.50	1.29	1.63	1.16	1.54	1.41	1.87	2.42	2.15
10	4.14	2.81	3.13	3.70	3.45	3.10	4.69	2.68	3.21	3.42	4.42	5.02	4.72
5	9.86	9.85	9.75	10.95	10.10	9.36	10.58	9.25	10.89	10.02	10.10	12.15	11.12
0	29.60	36.31	41.34	30.89	34.53	30.98	37.73	32.51	39.86	35.27	31.07	35.25	33.16
-5	67.42	76.57	81.99	65.44	72.86	71.11	75.42	72.71	79.98	74.81	68.10	75.85	71.97
Average	9.24	10.22	11.18	9.66	10.07	9.11	11.13	9.28	11.21	10.18	9.69	11.22	10.46
Multi condition													
	A										C		
SN	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway	Street	Average
∞	0.71	0.79	0.92	0.68	0.78	0.71	0.79	0.92	0.68	0.78	0.64	0.73	0.69
20	0.83	0.76	0.81	0.86	0.82	1.04	1.45	1.34	0.93	1.19	2.18	2.09	2.14
15	1.14	1.27	1.13	1.30	1.21	2.33	1.72	1.67	1.64	1.84	2.27	2.21	2.24
10	2.30	2.42	2.30	2.59	2.40	4.30	3.51	3.01	3.27	3.52	2.67	3.23	2.95
5	5.25	6.02	5.61	6.51	5.85	9.33	6.86	8.05	8.98	8.30	5.34	7.35	6.34
0	17.41	26.54	27.86	19.38	22.80	25.85	25.03	24.93	30.61	26.61	15.84	22.40	19.12
-5	49.40	65.66	72.59	46.96	58.65	65.83	60.46	63.47	71.95	65.43	47.47	59.58	53.52
Average	5.39	7.40	7.54	6.13	6.61	8.57	7.71	7.80	9.09	8.29	5.66	7.46	6.56

TABLE IV
SUMMARY OF ALL RESULTS BY PROPOSED METHOD WHICH USE DRW BY LDA-GMM OF $\boldsymbol{d}^0$ FOR THE REGION-WEIGHTING STEP, $\boldsymbol{A}_k[1\ \boldsymbol{d}^{4\top}]^\top (\lambda = 10^{-3})$ FOR THE TRANSFORMATION STEP (#11 IN TABLE II) FOR AURORA 2.

Clean condition													
	A					B					C		
SN	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway	Street	Average
∞	0.77	1.09	1.25	0.77	0.97	0.77	1.09	1.25	0.77	0.97	1.04	1.33	1.19
20	0.61	0.60	0.78	1.45	0.86	0.71	0.88	0.75	0.43	0.69	0.61	0.97	0.79
15	1.11	0.91	1.22	1.67	1.23	0.77	1.15	0.69	0.89	0.88	0.98	1.45	1.22
10	1.81	2.06	2.48	2.99	2.34	2.15	2.81	1.91	2.41	2.32	2.64	3.99	3.31
5	5.74	7.13	7.75	7.16	6.94	6.60	7.80	6.41	8.24	7.26	8.78	12.52	10.65
0	20.33	27.63	28.42	23.23	24.90	23.18	30.08	22.19	30.33	26.45	29.08	40.11	34.59
-5	53.48	66.54	69.79	56.09	61.47	62.63	68.86	61.23	70.75	65.87	65.67	74.85	70.26
Average	5.92	7.67	8.13	7.30	7.25	6.68	8.54	6.39	8.46	7.52	8.42	11.81	10.11
Multi condition													
	A					B					C		
SN	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway	Street	Average
∞	0.64	0.57	0.78	0.62	0.65	0.64	0.57	0.78	0.62	0.65	0.68	0.60	0.64
20	0.71	0.60	0.89	1.36	0.89	0.71	0.94	0.95	0.77	0.84	0.74	0.94	0.84
15	0.89	0.88	0.78	1.45	1.00	1.35	1.18	1.28	1.42	1.31	1.11	1.63	1.37
10	1.50	1.81	1.94	2.78	2.01	2.67	2.21	2.65	3.39	2.73	2.09	3.26	2.68
5	4.24	5.99	5.79	5.43	5.36	7.06	6.23	7.61	9.04	7.49	5.80	9.28	7.54
0	13.79	23.00	21.44	16.91	18.79	23.92	24.00	20.61	28.17	24.18	20.02	30.93	25.48
-5	42.92	63.75	65.20	47.82	54.92	64.51	60.10	59.44	68.84	63.22	57.23	69.01	63.12
Average	4.23	6.46	6.17	5.59	5.61	7.14	6.91	6.62	8.56	7.31	5.95	9.21	7.58

in both the clean and multi-style conditions. This may indicate that the proposed feature enhancement method makes full use of multi-style training data and that there is little room for performance improvement by multi-style acoustic model training.

In our preliminary experiments, we found that the performance was seriously degraded when we used Principal Component Analysis (PCA) instead of LDA. The dimensionality reduction matrix of PCA differed significantly from that of LDA. This indicates the importance of the use of discriminative criteria for dimensionality reduction of the joint feature vector space.

The relatively poor performance obtained by the proposed method in set C can be attributed to the fact that neither our method nor the noise estimation method of [19] used for our experiment take convolutional noise into account, which should be addressed in the future.

Tables III and IV show word accuracy details for NMN-SPLICE and the proposed method with the setups that achieved the highest word accuracies for each approach in the multi-style condition as shown in Table II, i.e., #5, and #11 for NMN-SPLICE and the proposed method, respectively. Our proposed method consistently outperformed NMN-SPLICE in all noise types and almost all SNR conditions in both sets A and B. These results show the robustness of the proposed approach with regard to changes in noise environments. In summary, the proposed method demonstrated an average WER improvement of 22.2% in the clean condition and 8.1% in the multi-style condition relative to NMN-SPLICE.

In addition to the above described experiments, we investigated performance sensitivity to changes in λ. Table V shows the average WERs that were obtained when we varied the λ

TABLE V
AVERAGE WERS (%) OBTAINED WITH VARIOUS λ.

λ	Set A	Set B	Set C	Ave.
10^{-1}	5.78	8.08	8.23	7.20
10^{-3} (#11)	**5.61**	7.31	**7.58**	**6.68**
10^{-5}	5.87	**7.10**	7.70	6.73
10^{-7}	5.91	7.13	7.92	6.80

TABLE VI
AVERAGE WERS (%) AND REAL-TIME FACTORS (RTFS) OBTAINED BY THE PROPOSED METHODS WITH VARIOUS N^T AND VTS ENHANCEMENT.

	Set A	Set B	Set C	Ave.	RTF
$N^T = 0$, $\lambda = 0$ (#10)	6.54	8.20	8.62	7.62	0.14
$N^T = 2$, $\lambda = 10^{-3}$	5.62	7.39	7.59	6.79	0.19
$N^T = 4$, $\lambda = 10^{-3}$ (#11)	**5.61**	**7.31**	**7.58**	**6.68**	0.21
$N^T = 6$, $\lambda = 10^{-3}$	5.85	7.25	8.03	6.85	0.25
VTS enhancement	9.80	11.91	9.93	10.67	0.45

value using the same configuration as the same configuration as condition #11 of Table II. The results in Table V show that although $\lambda = 10^{-3}$ was optimal among those considered, the proposed method was not very sensitive to the change in the λ value.

Finally, we compared our proposed methods with VTS enhancement using time-varying noise feature estimates, evaluating WERs and computational complexity. For the proposed method, we varied the N^T value to assess how different choices of N^T affected the resultant performance. As for VTS enhancement, we used the time-varying estimates of noise feature vectors that were used for the proposed methods. To keep computational complexity of VTS enhancement comparable to that of the proposed methods, it was performed in the static MFCC domain by using a 32-component GMM. Table VI shows the average WERs and Real-Time Factors (RTFs) obtained by the proposed methods with various N^T (we did not use the information from neighboring frames for the region-weighting step) and VTS enhancement in the multi-style condition. The result with $N^T = 0$ and $N^T = 4$ were identical, respectively, to those of #10 and #11 in Table II. The RTFs were measured on an Intel(R) Xeon(R) 3.00 GHz without parallelization. Comparing the proposed method with the VTS enhancement method, the former achieved lower WERs with smaller RTFs. Although using a longer acoustic context required more computational cost, the extra cost was marginal. The smallest WER was achieved by $N^T = 4$ (#11 in Table II).

V. CONCLUSIONS

In this paper we proposed the concept of DRW for achieving high speech recognition performance in a variety of noise environments with feasible computational cost. DRW estimates weights for partitioning the feature vector space by using a pre-trained model that is discriminatively trained using index posterior of a clean GMM as a label. In addition, we proposed utilizing temporally adjacent segments of corrupted and noise features in order to leverage dynamic characteristics of feature vectors. Experimental results using Aurora 2 confirmed the effectiveness of our proposed method. Our future work includes making the proposed method robust against convolutional noise and integrating the proposed method with advanced techniques such as discriminative training of linear transformations [21] and uncertainty decoding [26] for further performance improvement.

REFERENCES

[1] J. Droppo and A. Acero, "Environmental robustness," in *Handbook of Speech Processing*, J. Benesty, M. Sondhi, and Y. Huang, Eds. New York, NY, USA: Springer, 2007, pp. 653–679.
[2] *Robust Speech Recognition of Uncertain or Missing Data*, D. Kolossa and R. Haeb-Umbach, Eds. New York, NY, USA: Springer, 2011.
[3] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *J. Acoust. Soc. Amer.*, vol. 55, p. 1304, 1974.
[4] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 578–589, Oct. 1994.
[5] A. de la Torre, A. M. Peinado, J. C. Segura, J. L. Perez-Cordoba, M. C. Benitez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 355–366, May 2005.
[6] M. Duncan, M. Laurent, N. Bernhard, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," in *Proc. ICSLP*, 2002, pp. 17–20.
[7] J. Droppo, L. Deng, and A. Acero, "Evaluation of SPLICE on the Aurora 2 and 3 tasks," in *Proc. ICSLP*, 2002, pp. 29–32.
[8] 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," in *Proc. EUROSPEECH Experiments using the Aurora II Database and Tasks*, 2001, pp. 221–224.
[9] V. Stouten, "Robust automatic speech recognition in time-varying environments," Ph.D. dissertation, K. U. Leuven, Leuven, Belgium, 2006.
[10] J. Du and Q. Huo, "A feature compensation approach using high-order vector Taylor series approximation of an explicit distortion model for noisy speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 8, pp. 2285–2293, Nov. 2011.
[11] M. J. F. Gales and S. J. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Trans. Speech Audio Process.*, vol. 4, no. 5, pp. 352–359, Sep. 1996.
[12] A. Acero, L. Deng, T. Kristjansson, and J. Zhang, "HMM adaptation using vector Taylor series for noisy speech recognition," in *Proc. ICSLP*, 2000, pp. 869–872.
[13] Y. Zhao and B. H. Juang, "On noise estimation for robust speech recognition using vector Taylor series," in *Proc. ICASSP*, 2010, pp. 4290–4293.
[14] M. Fujimoto, S. Watanabe, and T. Nakatani, "Noise suppression with unsupervised joint speaker adaptation and noise mixture model estimation," in *Proc. ICASSP*, Mar. 2012, pp. 4713–4716.
[15] H. Liao and M. J. F. Gales, "Issues with uncertainty decoding for noise robust automatic speech recognition," *Speech Commun.*, vol. 50, no. 4, pp. 265–277, Apr. 2008.
[16] T. Kai, M. Suzuki, K. Chijiiwa, N. Minematsu, and K. Hirose, "Combination of SPLICE and feature normalization for noise robust speech recognition," *J. Signal Process.*, vol. 16, no. 4, pp. 323–326, Jul. 2012.
[17] Y. Qiao and N. Minematsu, "Mixture of probabilistic linear regressions: A unified view of GMM-based mapping techniques," in *Proc. ICASSP*, 2009, pp. 3913–3916.
[18] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 5, pp. 466–475, Sep. 2003.
[19] T. Yoshioka and T. Nakatani, "Speech enhancement based on log spectral envelope model and harmonicity-derived spectral mask, and its coupling with feature compensation," in *Proc. ICASSP*, 2011, pp. 5064–5067.
[20] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig, "fMPE: Discriminatively trained features for speech recognition," in *Proc. ICASSP*, 2005, vol. 1, pp. 961–964.
[21] J. Droppo, M. Mahajan, A. Gunawardana, and A. Acero, "How to train a discriminative front end with stochastic gradient descent and maximum mutual information," in *Proc. ASRU*, 2005, pp. 41–46.
[22] B. Zhang, S. Matsoukas, and R. Schwartz, "Discriminatively trained region dependent feature transforms for speech recognition," in *Proc. ICASSP*, 2006, vol. 1, pp. 313–316.
[23] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using iterative stochastic approximation for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 6, pp. 568–580, Nov. 2003.
[24] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR*, 2000.

[25] D. Pierce and A. Gunawardana, "Aurora 2.0 speech recognition in noise: Update 2," *Complex Backend Definition for Aurora 2.0*, 2002.
[26] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. ICASSP*, 2002, pp. 221–224.

Masayuki Suzuki received the B.Eng., M.Eng., Ph.D. degrees in electrical engineering and information systems from the University of Tokyo, Tokyo, Japan, in 2008, 2010, and 2013, respectively. From 2013, he has been working at IBM Research – Tokyo, Tokyo, Japan. His research interests include speech and spoken language processing, and pattern recognition. He is a member of the Acoustical Society of Japan (ASJ), the Institute of Electronics, Information and Communications Engineers (IEICE), IEEE, and ISCA. He received the Awaya Award from the ASJ in 2013.

Takuya Yoshioka received the B.Eng., M.Inf., and Ph.D. degrees in informatics from Kyoto University, Kyoto, Japan, in 2004, 2006, and 2010, respectively.

He is a Researcher at NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan, and is currently a Visiting Scholar at the Machine Intelligence Laboratory, the University of Cambridge, Cambridge, UK. In 2005, he interned at NTT, where he conducted research of dereverberation. Since joining NTT in 2006, he has been working on the development of algorithms for noise robust speech recognition, speech enhancement, and microphone arrays.

Dr. Yoshioka received the Awaya Prize Young Researcher Award and the Itakura Prize Innovative Young Researcher Award from the Acoustical Society of Japan (ASJ) in 2010 and 2011, respectively, and the Young Researcher's Award in Speech Field from the Institute of Electronics, Information and Communication Engineers (IEICE) Information and Systems Society (ISS) in 2011.

Shinji Watanabe received his B.S., M.S., and Dr. Eng. degrees from Waseda University, Tokyo, Japan, in 1999, 2001, and 2006, respectively. From 2001 to 2011, he was a research scientist at NTT Communication Science Laboratories, Kyoto, Japan. From January to March in 2009, he was a visiting scholar in the Georgia Institute of Technology, Atlanta, GA. Since 2011, he has been working at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA. His research interests include Bayesian learning, pattern recognition, and speech and spoken language processing. He is a member of the Acoustical Society of Japan (ASJ), the Institute of Electronics, Information and Communications Engineers (IEICE), and IEEE. He received the Awaya Award from the ASJ in 2003, the Paper Award from the IEICE in 2004, the Itakura Award from ASJ in 2006, and the TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2006. He is currently an Associate Editor of the IEEE TRANSACTIONS ON AUDIO SPEECH AND LANGUAGE PROCESSING.

Nobuaki Minematsu received the Ph.D. degree in electronic engineering in 1995 from the University of Tokyo. In 1995, he became an assistant researcher with the Department of Information and Computer Science, Toyohashi University of Technology, and in 2000, he was an associate professor with the Graduate School of Engineering, the University of Tokyo. Since 2012, he has been a professor with the Graduate School of Engineering, the University of Tokyo. From 2002 to 2003, he was a visiting researcher at Kungl Tekniska Högskolan (KTH), Sweden. He has a wide interest in speech from science to engineering, including phonetics, phonology, language learning, speech perception, speech analysis, speech recognition, speech synthesis, and speech applications. Dr. Minematsu is a member of IEEE, ISCA, IPA, the Institute of Electronics, Information and Communication Engineering, the Acoustical Society of Japan, the Information Processing Society of Japan, the Japanese Society for Artificial Intelligence, and the Phonetic Society of Japan. He received best paper awards from Research Institute of Signal Processing in 2007 and 2013.

Keikichi Hirose (M'78-SM'12) received the B. E. degree in electrical engineering in 1972, and the M. E. and Ph. D. degrees in electronic engineering respectively in 1974 and 1977 from the University of Tokyo. From 1977, he is a faculty member at the University of Tokyo, and was a Professor of the Department of Electronic Engineering from 1994. Currently he is professor at the Department of Information and Communication Engineering, Graduate School of Information Science and Technology, University of Tokyo. From March 1987 to January 1988, he was Visiting Scientist at the Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, U.S.A. He has been engaged in a wide range of research on spoken language processing, including analysis, synthesis, recognition, dialogue systems, and computer-assisted language learning. From 2000 to 2004, he was Principal Investigator of the national project "Realization of advanced spoken language information processing utilizing prosodic features," supported by the Japanese Government. He served as the general chair for INTERSPEECH 2010, Makuhari, Japan. Since 2010, he serves as the Chair of ISCA Special Interest Group on Speech Prosody (SProSIG). He is a Fellow of IEICE, and a member of a number of academic societies, including International Speech Communication Association (Board member), Acoustical Society of America, ASJ, Information Processing Society of Japan, Japanese Society for Artificial Intelligence, and Research Institute of Signal Processing Japan (Board member).