STEREO-BASED STOCHASTIC NOISE COMPENSATION BASED ON TRAJECTORY GMMS

Heiga Zen, Yoshihiko Nankaku, Keiichi Tokuda

Department of Computer Science, Nagoya Institute of Technology, Nagoya, 466-8555 JAPAN
{zen,nankaku,tokuda}@sp.nitech.ac.jp

ABSTRACT

This paper proposes a novel stereo-based stochastic noise compensation technique based on trajectory GMMs. Although the GMM-based noise compensation techniques such as SPLICE work effective, their performance sometimes degrades due to the inappropriate dynamic characteristics caused by the frame-by-frame mapping. While the use of dynamic feature constraints on the mapping stage can alleviate this problem, it also introduces an inconsistency between training and mapping. The recently proposed trajectory GMM-based feature mapping technique can solve this inconsistency while keeping the benefits of the use of dynamic features, and offers an entire sequence-level transformation rather than the frame-by-frame mapping. Results from a noise compensation experiment on the AURORA-2 task show that the proposed trajectory GMM-based noise compensation technique outperforms the conventional ones.

Index Terms— speech recognition, GMM-based mapping, trajectory GMM, dynamic features, noise compensation

1. INTRODUCTION

During these last years, stereo-based continuous stochastic feature mapping techniques based on Gaussian mixture models (GMMs) have been applied to noise compensation [1, 2]. This mapping technique consists of three major stages:

1. Joint probability density functions (PDF) between noisy (source) and clean (target) features are modeled by a set GMMs using two-channel (so-called "stereo") data;
2. Conditional PDFs of clean features for given noisy ones are estimated from the joint PDFs;
3. The mapped clean features are determined so as to minimize their mean-square error (MMSE) from the conditional PDFs [2];

Although this technique makes it possible to continuously transform of any sample of the source into that of target, its mapping performance is still insufficient. One of major factors which deteriorate the mapping quality is inappropriate dynamic characteristics caused by the frame-by-frame mapping. To alleviate this problem, GMM-based feature mapping technique with dynamic feature constraints have been proposed in voice conversion area [3]. The use of dynamic features makes it possible to convert a source feature vector sequence to a target one while satisfying the statistics of both static and dynamic features. However, it also introduces an inconsistency between training and mapping [4]. Generally, dynamic features are calculated as regression coefficients from static features of their neighboring frames. Thus, the relationship between the static and dynamic features is deterministic. In the conventional mapping techniques with dynamic features, this relationship is ignored on both the joint PDF training and conditional PDF estimation stages but utilized on the mapping stage [3].

Recently, a trajectory model, derived from the HMM by imposing the explicit relationships between static and dynamic features, was proposed [4]. This model, named trajectory HMM, can overcome the frame-wise conditional independence assumption of state output probabilities and piecewise constant statistics of the HMM, without any additional parameters. Based on this idea, Zen et al. also proposed the continuous stochastic feature mapping technique based on trajectory GMMs [5], i.e. joint PDFs between source and target feature vectors are modeled by a set of trajectory GMMs and the conditional PDFs are estimated from them. This technique can solve the inconsistency introduced by the use of dynamic features while keeping its benefits, and offers an entire sequence-level transformation rather than the frame-by-frame mapping. In this paper, we apply this new mapping technique to noise compensation.

This paper is organized as follows. Section 2 describes the mapping technique based on trajectory GMMs. Section 3 shows experimental results in noise compensation on the AURORA-2 task. Concluding remarks and future plans are given in the final section.

2. NOISE COMPENSATION BASED ON TRAJECTORY GMMS

2.1. Joint Probability Density Function

Noisy and clean MFCC vector sequences, $\boldsymbol{x}$ and $\boldsymbol{y}$, are written as

$$\boldsymbol{x} = \left[\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_T^\top\right]^\top, \quad \boldsymbol{y} = \left[\boldsymbol{y}_1^\top, \ldots, \boldsymbol{y}_T^\top\right]^\top, \tag{1}$$

where $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ are the M-dimensional noisy and clean MFCC static feature vectors at the t-th frame, respectively, and T is the total number of frames. In this paper, the joint probability of $\boldsymbol{x}$ and $\boldsymbol{y}$ is modeled by a trajectory GMM [5] as follows:

$$p\left(\boldsymbol{z} \mid \lambda\right) = \sum_{\forall \boldsymbol{q}} P\left(\boldsymbol{q} \mid \lambda\right) p\left(\boldsymbol{z} \mid \boldsymbol{q}, \lambda\right), \tag{2}$$

$$p\left(\boldsymbol{z} \mid \boldsymbol{q}, \lambda\right) = \mathcal{N}\left(\boldsymbol{z}\,;\, \bar{\boldsymbol{z}}_{\boldsymbol{q}}, \boldsymbol{P}_{\boldsymbol{q}}\right), \tag{3}$$

$$P\left(\boldsymbol{q} \mid \lambda\right) = \prod_{t=1}^{T} c_{q_t}, \tag{4}$$

$$\boldsymbol{z} = \left[\boldsymbol{z}_1^\top, \ldots, \boldsymbol{z}_T^\top\right]^\top, \quad \boldsymbol{z}_t = \left[\boldsymbol{x}_t^\top, \boldsymbol{y}_t^\top\right]^\top, \tag{5}$$

where λ denotes the set of model parameters, $\boldsymbol{z}$ is an $2MT \times 1$ joint MFCC vector sequence, $\boldsymbol{z}_t$ is the $2M \times 1$ joint MFCC vector at the t-th frame, $\boldsymbol{q} = \{q_1, \ldots, q_T\}$ is a Gaussian component sequence, $q_t \in$

Heiga Zen is now with the Speech Technology Group at Toshiba Research Europe Ltd. Cambridge Research Laboratory in Cambridge, CB4 0GZ, UK.

$\{1, \ldots, N\}$ is the Gaussian component at the t-th frame, N is the total number of Gaussian components in the model set, and c_i is the mixture prior probability of the i-th Gaussian component. In Eq. (3), $\bar{\boldsymbol{z}}_{\boldsymbol{q}}$ and $\boldsymbol{P}_{\boldsymbol{q}}$ are the $2MT \times 1$ mean vector and the $2MT \times 2MT$ covariance matrix for $\boldsymbol{q}$, respectively. They are given by

$$\boldsymbol{R}_{\boldsymbol{q}} \bar{\boldsymbol{z}}_{\boldsymbol{q}} = \boldsymbol{r}_{\boldsymbol{q}}, \tag{6}$$

$$\boldsymbol{R}_{\boldsymbol{q}} = \boldsymbol{W}^{\top} \boldsymbol{\Omega}_{\boldsymbol{q}} \boldsymbol{W} = \boldsymbol{P}_{\boldsymbol{q}}^{-1}, \tag{7}$$

$$\boldsymbol{r}_{\boldsymbol{q}} = \boldsymbol{W}^{\top} \boldsymbol{\Omega}_{\boldsymbol{q}} \boldsymbol{\mu}_{\boldsymbol{q}}, \tag{8}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}} = \left[\boldsymbol{\mu}_{q_1}^{\top}, \ldots, \boldsymbol{\mu}_{q_T}^{\top} \right]^{\top}, \tag{9}$$

$$\boldsymbol{\mu}_i = \left[\boldsymbol{\mu}_i^{(\boldsymbol{x})\top}, \boldsymbol{\mu}_i^{(\boldsymbol{y})\top} \right]^{\top}, \tag{10}$$

$$\boldsymbol{\Omega}_{\boldsymbol{q}} = \mathrm{diag}\left[\boldsymbol{\Omega}_{q_1}, \ldots, \boldsymbol{\Omega}_{q_T} \right], \tag{11}$$

$$\boldsymbol{\Omega}_i = \begin{bmatrix} \boldsymbol{\Omega}_i^{(\boldsymbol{xx})} & \boldsymbol{\Omega}_i^{(\boldsymbol{xy})} \\ \boldsymbol{\Omega}_i^{(\boldsymbol{yx})} & \boldsymbol{\Omega}_i^{(\boldsymbol{yy})} \end{bmatrix}, \tag{12}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\Omega}_i$ are the $6M \times 1$ mean vector and the $6M \times 6M$ precision (inverse covariance) matrix associated with the i-th Gaussian component, respectively, and $\boldsymbol{W}$ is a $6MT \times 2MT$ window matrix which appends dynamic features to $\boldsymbol{z}$. Unfortunately, estimating this model based on the ML criterion via the exact EM algorithm is computationally infeasible because it is intractable to compute $p(\boldsymbol{q} \mid \boldsymbol{z}, \lambda)$. However, approximate training technique based on the Viterbi approximation or Markov chain Monte Carlo (MCMC) has been proposed.

Equations (2)–(4) show that the trajectory GMM can be interpreted as an $2MT$-dimensional GMM whose mixture weights are given by products of the mixture prior probabilities. Note that the intra and inter-frame covariance matrix of Eq. (3), $\boldsymbol{P}_{\boldsymbol{q}}$, is generally full.Therefore, the trajectory GMM can model both intra and inter-frame dependencies between noisy and clean MFCC vector sequences without increasing the number of model parameters compared to the GMM with the same model topology.

2.2. Mapping

2.2.1. Conditional probability

Equation (3) can be rewritten as follows:

$$p\left(\boldsymbol{z} \mid \boldsymbol{q}, \lambda\right) = \mathcal{N}\left(\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} ; \begin{bmatrix} \bar{\boldsymbol{x}}_{\boldsymbol{q}} \\ \bar{\boldsymbol{y}}_{\boldsymbol{q}} \end{bmatrix}, \begin{bmatrix} \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{xx})} & \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{xy})} \\ \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yx})} & \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yy})} \end{bmatrix} \right), \tag{13}$$

where

$$\boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{xx})} = \boldsymbol{C}_{\boldsymbol{q}}^{(\boldsymbol{xx})^{-1}}, \quad \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yy})} = \boldsymbol{C}_{\boldsymbol{q}}^{(\boldsymbol{yy})^{-1}}, \tag{14}$$

$$\boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{xy})} = -\boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xx})^{-1}} \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xy})} \boldsymbol{C}_{\boldsymbol{q}}^{(\boldsymbol{yy})^{-1}} = \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yx})^{\top}}, \tag{15}$$

$$\boldsymbol{C}_{\boldsymbol{q}}^{(\boldsymbol{xx})} = \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xx})} - \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xy})} \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yy})^{-1}} \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yx})}, \tag{16}$$

$$\boldsymbol{C}_{\boldsymbol{q}}^{(\boldsymbol{yy})} = \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yy})} - \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yx})} \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xx})^{-1}} \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xy})}, \tag{17}$$

$$\boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xx})} = \boldsymbol{W}^{(\boldsymbol{x})^{\top}} \boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{xx})} \boldsymbol{W}^{(\boldsymbol{x})}, \tag{18}$$

$$\boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yy})} = \boldsymbol{W}^{(\boldsymbol{y})^{\top}} \boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{yy})} \boldsymbol{W}^{(\boldsymbol{y})}, \tag{19}$$

$$\boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xy})} = \boldsymbol{W}^{(\boldsymbol{x})^{\top}} \boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{xy})} \boldsymbol{W}^{(\boldsymbol{y})} = \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yx})^{\top}}, \tag{20}$$

$$\boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{xx})} = \mathrm{diag}\left[\boldsymbol{\Omega}_{q_1}^{(\boldsymbol{xx})}, \ldots, \boldsymbol{\Omega}_{q_T}^{(\boldsymbol{xx})} \right], \tag{21}$$

$$\boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{yy})} = \mathrm{diag}\left[\boldsymbol{\Omega}_{q_1}^{(\boldsymbol{yy})}, \ldots, \boldsymbol{\Omega}_{q_T}^{(\boldsymbol{yy})} \right], \tag{22}$$

$$\boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{xy})} = \mathrm{diag}\left[\boldsymbol{\Omega}_{q_1}^{(\boldsymbol{xy})}, \ldots, \boldsymbol{\Omega}_{q_T}^{(\boldsymbol{xy})} \right] = \boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{yx})}, \tag{23}$$

and $\boldsymbol{W}^{(\boldsymbol{x})} = \boldsymbol{W}^{(\boldsymbol{y})}$ is the $3MT \times MT$ window matrix which appends dynamic features to $\boldsymbol{x}$ and $\boldsymbol{y}$. The mean vectors of Eq. (13), $\bar{\boldsymbol{x}}_{\boldsymbol{q}}$ and $\bar{\boldsymbol{y}}_{\boldsymbol{q}}$, are given as follows:

$$\bar{\boldsymbol{x}}_{\boldsymbol{q}} = \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{xx})} \left(\boldsymbol{r}_{\boldsymbol{q}}^{(\boldsymbol{x})} - \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xy})} \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yy})^{-1}} \boldsymbol{r}_{\boldsymbol{q}}^{(\boldsymbol{y})} \right), \tag{24}$$

$$\bar{\boldsymbol{y}}_{\boldsymbol{q}} = \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yy})} \left(\boldsymbol{r}_{\boldsymbol{q}}^{(\boldsymbol{y})} - \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{yx})} \boldsymbol{R}_{\boldsymbol{q}}^{(\boldsymbol{xx})^{-1}} \boldsymbol{r}_{\boldsymbol{q}}^{(\boldsymbol{x})} \right), \tag{25}$$

where

$$\boldsymbol{r}_{\boldsymbol{q}}^{(\boldsymbol{x})} = \boldsymbol{W}^{(\boldsymbol{x})^{\top}} \left(\boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{xx})} \boldsymbol{\mu}_{\boldsymbol{q}}^{(\boldsymbol{x})} + \boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{xy})} \boldsymbol{\mu}_{\boldsymbol{q}}^{(\boldsymbol{y})} \right), \tag{26}$$

$$\boldsymbol{r}_{\boldsymbol{q}}^{(\boldsymbol{y})} = \boldsymbol{W}^{(\boldsymbol{y})^{\top}} \left(\boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{yy})} \boldsymbol{\mu}_{\boldsymbol{q}}^{(\boldsymbol{y})} + \boldsymbol{\Omega}_{\boldsymbol{q}}^{(\boldsymbol{yx})} \boldsymbol{\mu}_{\boldsymbol{q}}^{(\boldsymbol{x})} \right), \tag{27}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}}^{(\boldsymbol{x})} = \left[\boldsymbol{\mu}_{q_1}^{(\boldsymbol{x})^{\top}}, \ldots, \boldsymbol{\mu}_{q_T}^{(\boldsymbol{x})^{\top}} \right]^{\top}, \tag{28}$$

$$\boldsymbol{\mu}_{\boldsymbol{q}}^{(\boldsymbol{y})} = \left[\boldsymbol{\mu}_{q_1}^{(\boldsymbol{y})^{\top}}, \ldots, \boldsymbol{\mu}_{q_T}^{(\boldsymbol{y})^{\top}} \right]^{\top}. \tag{29}$$

As a result, the conditional PDF of $\boldsymbol{y}$ for given $\boldsymbol{x}$ and λ can be expressed as

$$p\left(\boldsymbol{y} \mid \boldsymbol{x}, \lambda\right) = \sum_{\forall \boldsymbol{q}} \gamma_{\boldsymbol{q}} \cdot p\left(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{q}, \lambda\right), \tag{30}$$

$$p\left(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{q}, \lambda\right) = \mathcal{N}\left(\boldsymbol{y} ; \tilde{\boldsymbol{y}}_{\boldsymbol{q}}, \tilde{\boldsymbol{P}}_{\boldsymbol{q}}^{(\boldsymbol{yy})} \right), \tag{31}$$

$$\tilde{\boldsymbol{y}}_{\boldsymbol{q}} = \bar{\boldsymbol{y}}_{\boldsymbol{q}} + \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yx})} \boldsymbol{C}_{\boldsymbol{q}}^{(\boldsymbol{xx})} \left(\boldsymbol{x} - \bar{\boldsymbol{x}}_{\boldsymbol{q}} \right), \tag{32}$$

$$\tilde{\boldsymbol{P}}_{\boldsymbol{q}}^{(\boldsymbol{yy})} = \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yy})} - \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{yx})} \boldsymbol{C}_{\boldsymbol{q}}^{(\boldsymbol{xx})} \boldsymbol{P}_{\boldsymbol{q}}^{(\boldsymbol{xy})}, \tag{33}$$

where $\gamma_{\boldsymbol{q}}$ is a posterior probability of $\boldsymbol{q}$.

2.2.2. MMSE-based mapping

The MMSE estimates of the clean static MFCC vector sequence $\hat{\boldsymbol{y}}$ is determined as follows:

$$\hat{\boldsymbol{y}} = E\left[\boldsymbol{y} \mid \boldsymbol{x}\right] = \int p\left(\boldsymbol{y} \mid \boldsymbol{x}, \lambda\right) \boldsymbol{y} d\boldsymbol{y} \tag{34}$$

$$= \int \sum_{\forall \boldsymbol{q}} \gamma_{\boldsymbol{q}} \cdot p\left(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{q}, \lambda\right) \boldsymbol{y} d\boldsymbol{y} \tag{35}$$

$$= \sum_{\forall \boldsymbol{q}} \gamma_{\boldsymbol{q}} \cdot \tilde{\boldsymbol{y}}_{\boldsymbol{q}}, \tag{36}$$

where $E[\cdot]$ means expectation. The estimated MFCC static feature vector sequence $\hat{\boldsymbol{y}}$ is defined as the weighted sum of the conditional mean vectors $\tilde{\boldsymbol{y}}_{\boldsymbol{q}}$, where the posterior probabilities of $\boldsymbol{q}$ are used as weights.

Unfortunately, computing $\gamma_{\boldsymbol{q}}$ is computationally intractable. Hence, we also need to use the Viterbi or MCMC approximation like the ML estimation of trajectory GMMs. In the all experiments described in this paper, we approximated the marginalization over all possible $\boldsymbol{q}$ by a single $\hat{\boldsymbol{q}}$ and did not iterate the EM algorithm.

3. EXPERIMENTS

3.1. Experimental conditions

Speech recognition experiments were conducted under various noise conditions using the Aurora-2 database and task [6]. The Aurora-2 database is a subset of the TI-DIGITS, which contains a set of connected digit utterances spoken in English; while the task consists of the recognition of the connected digit utterances interfered with various noise sources at different signal-to-noise ratios (SNRs), in which the test-sets A (seen noise types) and B (unseen noise types) are artificially contaminated with eight different types of real world noises in a wide range of SNRs (-5dB, 0dB, 5dB, 10dB, 15dB, 20dB, and Clean) and the test-set C additionally includes the channel distortion. The acoustic model for each digit was a left-to-right continuous density HMM with 16 states, and each state had a mixture of three Gaussian distributions. Two additional silence models were defined. One had three states with a mixture of six Gaussian distributions per state for modeling the silence at the beginning and at the end of each utterance. The other one had one state with a mixture of six Gaussian distributions for modeling the inter-word short pause. A 39-dimensional feature vector was composed at each frame, including 12 MFCCs and the logarithm of the energy extracted by the reference Aurora front-end version 2.0, as well as their corresponding dynamic coefficients. The training and recognition tests used HTK, which followed the setup originally defined for the ETSI evaluations. All the experimental results reported below are based on clean-condition training, i.e., the acoustic models were trained only with the clean (uncontaminated) training utterances. The clean acoustic model training data consists of 8,440 utterances. The multi-style acoustic model training data was used to train mapping functions for noise compensation. It consists of the same utterances synthetically mixed with four different noise types (Subway, Car, Babble, and Exhibition) at varying amplitudes (5dB, 10dB, 15dB, 20dB, and Clean), for a total of 17 unique noise conditions. Using the pairs of clean and noisy speech data, joint static feature vectors consisted of pairs of clean and noisy MFCCs were composed. They were augmented with their dynamic features (78 dimensions in total)

In this experiment, four types of mapping techniques were evaluated, which were

- `GMM-Static`: The GMM-based mapping [2] was applied to the 13-dimensional static MFCC vectors;
- `GMM-Complete`: Like Complete SPLICE [7], the 13-dimensional static MFCC vectors were augmented with their dynamic features before passing them to the GMM-based mapping. Note that the dynamic features in mapped 39-dimensional vectors were no longer correspond to true dynamic features;
- `GMM-Dynamic`: The 13-dimensional static MFCC vectors were mapped by GMMs under the constraints between static and dynamic features [3];
- `Trajectory GMM`: The 13-dimensional static MFCC vectors were transformed by the trajectory GMM-based mapping;

All of the four mapping techniques used a mixture of 256 Gaussian components to model the joint probability of clean and noisy static MFCC vectors for each of the 17 conditions. The diagonal structure was used for $\mathbf{\Omega}_i^{(xx)}$, $\mathbf{\Omega}_i^{(xy)}$, $\mathbf{\Omega}_i^{(yx)}$, and $\mathbf{\Omega}_i^{(yy)}$. The GMMs for `GMM-Static` were estimated using 26-dimensional joint static feature vector sequences, and the others were estimated using 78-dimensional augmented ones (with dynamic features). Note that the GMMs for `GMM-Complete` and `GMM-Dynamic` were identical because the difference between these two techniques is the use of dynamic feature constraints at the mapping stage only. The trajectory GMMs were re-estimated by the Viterbi training based on the ML criterion [4] using the GMMs for `GMM-Dynamic` as their initial models.

To select the mapping function in the decoding stage, a GMM-based noise condition classifier was built for the detection of noise type and SNR. In this noise condition classifier, a GMM with four Gaussian components was estimated for each combination of noise type and SNR using the first 10 frames of the utterances from that noise condition in the training data. Before decoding, the likelihood of the first 10 frames of the input utterances was computed against each GMM. The noise condition with the maximum likelihood was chosen as the noise condition of the input utterance, and the mapping of that noise condition was applied to compensate the utterance unless the noise condition was Clean. The dynamic features of mapped static features by `GMM-Complete`, `GMM-Dynamic`, and `Trajectory GMM` were computed on-line during recognition.

3.2. Experimental results

Figure 1 is a summary of the full results on the Aurora-2 corpus. Note that the performance of the baseline system (clean condition training, without noise compensation) was the same as that of the ETSI reference system described in [6], and the average accuracy and relative error reduction were computed with the results between 20 and 0 dB as suggested in [6]. It can be seen from the figure that all of four mapping techniques achieved significant improvements over the baseline system. Compared with the other three techniques, the performance of `GMM-Static` was significantly worse. It demonstrates the effectiveness of augmenting static features by their dynamic features in noise compensation. Although the GMMs of `GMM-Dynamic` and `GMM-Complete` were identical, `GMM-Dynamic` slightly outperformed `GMM-Complete` on test-set A (seen noise types). On the other hand, on test-set B (unseen noise types), `GMM-Complete` achieved the lower WERs than `GMM-Dynamic`. Similar tendency can also be observed in `Trajectory GMM`: `Trajectory GMM` outperformed `GMM-Complete` on test-set A but `GMM-Complete` achieved the lower WERs than `Trajectory GMM` on test-set B. For seen noise types with unseen SNRs (SNR=0 or SNR=-5dB in test-set A), `GMM-Dynamic` and `Trajectory GMM` outperformed `GMM-Complete`. These results suggest that incorporating dynamic feature constraints in the mapping stage improves the mapping performance for seen noise types but degrades the recognition performance for unseen noise types. We expect that the dynamic feature constraints caused over-fitting to the training noise types. In total, `Trajectory GMM` achieved the lowest WERs among the four mapping techniques.

4. CONCLUSIONS

This paper evaluated the performance of stochastic feature mapping technique based on trajectory GMMs on noise compensation. Experimental results showed that the use of dynamic feature constraints in noise compensation worked effective in seen noise types. It also indicated that the use of dynamic features degraded the mapping performance due to over-fitting to the training data. In total, the proposed noise compensation algorithm outperformed the conventional mapping techniques on the AURORA-2 task.

Aurora 2 Clean Training + GMM-Static

	A					B					C			Overall	Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Average	Improvement
Clean	98.86	99.00	98.96	99.23	99.01	98.86	99.00	98.96	99.23	99.01	99.14	98.97	99.06	99.02	-0.56%
20 dB	97.33	98.10	97.70	97.53	97.67	96.99	96.46	96.93	96.91	96.82	96.75	95.01	95.88	96.97	37.63%
15 dB	94.60	95.98	96.27	95.62	95.62	95.00	92.59	93.14	94.35	93.77	93.06	89.81	91.44	94.04	51.53%
10 dB	90.27	90.81	90.75	91.51	90.84	87.38	83.07	85.09	85.75	85.32	81.27	77.45	79.36	86.34	56.58%
5 dB	80.50	72.61	77.27	80.13	77.63	69.63	59.31	63.14	64.83	64.23	54.31	53.66	53.99	67.54	45.45%
0 dB	59.16	35.70	46.82	61.06	50.69	37.70	29.14	28.96	29.99	31.45	25.64	26.03	25.84	38.02	25.02%
-5 dB	26.90	4.05	16.13	32.12	19.80	7.55	10.01	3.13	1.88	5.64	11.54	12.85	12.20	12.62	4.52%
Average	84.37	78.64	81.76	85.17	82.49	77.34	72.11	73.45	74.37	74.32	70.21	68.39	69.30	76.58	
	48.77%	57.38%	53.71%	57.15%	54.69%	52.20%	27.53%	43.21%	42.23%	41.96%	11.95%	6.73%	9.33%		41.36%

Aurora 2 Clean Training + GMM-Complete

	A					B					C			Overall	Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Average	Improvement
Clean	98.96	99.00	98.96	99.23	99.04	98.96	99.00	98.96	99.23	99.04	99.14	98.97	99.06	99.04	1.31%
20 dB	97.88	98.31	98.06	97.84	98.02	97.48	97.64	97.82	97.87	97.70	97.42	97.22	97.32	97.75	53.55%
15 dB	96.75	96.83	96.96	96.88	96.86	96.28	96.46	96.36	95.53	96.16	94.96	94.89	94.93	96.19	69.46%
10 dB	93.37	94.23	94.93	93.64	94.04	91.71	91.57	91.05	90.71	91.26	87.66	86.43	87.05	91.53	72.89%
5 dB	88.12	84.89	88.04	86.27	86.83	80.47	78.96	79.54	78.25	79.31	70.53	68.95	69.74	80.40	66.78%
0 dB	69.76	53.63	63.88	70.32	64.40	50.51	51.09	50.52	46.13	49.56	39.58	41.57	40.58	53.70	43.89%
-5 dB	32.98	18.65	23.83	32.34	26.95	14.92	18.71	12.44	11.82	14.47	17.75	19.50	18.63	20.29	12.85%
Average	89.18	85.58	88.37	88.99	88.03	83.29	83.14	83.06	81.70	82.80	78.03	77.81	77.92	83.92	
	64.52%	71.22%	70.49%	68.18%	69.03%	64.75%	56.20%	63.76%	58.75%	61.13%	35.07%	34.53%	34.80%		59.72%

Aurora 2 Clean Training + GMM-Dynamic

	A					B					C			Overall	Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Average	Improvement
Clean	98.93	99.00	98.96	99.23	99.03	98.93	99.00	98.96	99.23	99.03	99.14	98.97	99.06	99.04	0.75%
20 dB	98.16	98.19	98.27	98.21	98.21	97.45	97.49	98.03	97.22	97.55	97.54	97.37	97.46	97.79	55.32%
15 dB	96.65	97.19	97.55	96.95	97.09	95.67	95.77	95.76	94.57	95.44	95.39	94.44	94.92	95.99	68.34%
10 dB	94.11	94.59	95.59	93.98	94.57	90.08	90.15	90.01	89.76	90.00	89.32	86.91	88.12	91.45	73.22%
5 dB	88.73	84.85	88.70	86.58	87.22	78.26	77.09	77.54	75.29	77.05	75.65	70.56	73.11	80.33	67.11%
0 dB	71.48	52.24	65.05	68.13	64.23	46.27	46.70	46.47	41.10	45.14	43.75	42.20	42.98	52.34	42.43%
-5 dB	35.46	17.26	24.16	32.46	27.34	12.59	18.14	14.94	11.20	14.22	18.27	19.20	18.74	20.37	12.96%
Average	89.83	85.41	89.03	88.77	88.26	81.55	81.44	81.56	79.59	81.03	80.33	78.30	79.31	83.58	
	66.65%	70.89%	72.16%	67.55%	69.63%	61.07%	51.77%	60.56%	54.00%	57.14%	41.87%	35.95%	38.91%		58.88%

Aurora 2 Clean Training + Trajectory GMM

	A					B					C			Overall	Percentage
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	Average	Improvement
Clean	98.93	99.00	98.96	99.23	99.03	98.93	99.00	98.96	99.23	99.03	99.14	98.97	99.06	99.04	0.75%
20 dB	97.97	98.16	98.18	98.40	98.18	97.39	97.58	97.61	97.16	97.44	97.67	97.43	97.55	97.76	54.73%
15 dB	96.81	97.16	97.46	96.79	97.06	95.18	95.98	95.05	94.63	95.21	96.01	94.74	95.38	95.98	68.73%
10 dB	94.66	94.77	95.44	94.72	94.90	88.73	90.45	89.53	90.44	89.79	90.30	87.30	88.80	91.63	74.16%
5 dB	90.60	84.49	90.67	87.84	88.40	77.40	77.42	76.86	77.82	77.38	78.39	71.95	75.17	81.34	68.99%
0 dB	76.05	54.23	70.03	73.25	68.39	47.34	48.40	48.67	47.21	47.91	49.34	45.56	47.45	56.01	46.93%
-5 dB	41.57	15.72	30.15	39.46	31.73	12.19	17.17	13.48	11.94	13.70	20.02	20.56	20.29	22.23	15.05%
Average	91.22	85.76	90.36	90.20	89.38	81.21	81.97	81.54	81.45	81.54	82.34	79.40	80.87	84.54	
	71.21%	71.59%	75.52%	71.68%	72.54%	60.36%	53.14%	60.52%	58.20%	58.29%	47.81%	39.20%	43.50%		61.30%

Fig. 1. Summary of the full results by GMM and trajectory GMM-based noise compensation techniques on the AURORA-2 task.

5. ACKNOWLEDGMENTS

This work was partly supported by the MEXT e-Society project, the Hori information science promotion foundation, and the Grant-in-Aid for Scientific Research (No. 1880009) of JSPS.

6. REFERENCES

[1] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. ICASSP*, 2001, pp. 301–304.

[2] X.-D. Cui, M. Afify, and Y. Gao, "MMSE-based stereo feature stochastic mapping for noise robust speech recognition," in *Proc. ICASSP*, 2008, pp. 4077–4080.

[3] T. Toda, A.W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," *IEEE Trans. ASLP*, vol. 15, no. 8, pp. 2222–2235, 2007.

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

[5] H. Zen, Y. Nankaku, and K. Tokuda, "Probabilistic feature mapping based on trajectory HMMs," in *Proc. Interspeech*, 2008.

[6] H.G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR-2000*, 2000, pp. 181–188.

[7] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. ICASSP*, 2002, pp. 57–60.