BAYESIAN FEATURE ENHANCEMENT USING A MIXTURE OF UNSCENTED TRANSFORMATIONS FOR UNCERTAINTY DECODING OF NOISY SPEECH

Yusuke Shinohara and Masami Akamine

Corporate Research and Development Center, Toshiba Corporation
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
yusuke.shinohara@toshiba.co.jp

ABSTRACT

A new parameter estimation method for the Model-Based Feature Enhancement (MBFE) is presented. The conventional MBFE uses the vector Taylor series to calculate the parameters of non-linearly transformed distributions, though the linearization leads to a degraded performance. We use the unscented transformation to estimate the parameters, where a minimal number of samples propagated through the non-linear transformation are used. By avoiding the linearization, the parameters are estimated more accurately. Experimental results on Aurora2 show that the proposed method reduces the word error rate by 8.48% relatively, while the computational cost is just modestly higher, compared with the conventional MBFE.

Index Terms— Feature enhancement, unscented transformation, vector Taylor series, uncertainty decoding, noisy speech recognition,

1. INTRODUCTION

The performance of speech recognition systems degrades in the presence of noise, which is a major problem in utilizing those systems in adverse environments. Various techniques have been proposed to improve the noise-robustness of speech recognizers. These techniques can be classified into one of the two major approaches, namely the feature enhancement and the acoustic model adaptation. In this work, we pursue the feature enhancement approach due to its computational efficiency.

Among many feature enhancement algorithms proposed in the literature, the Model-Based Feature Enhancement (MBFE) [1] is known to be a powerful and theoretically sound technique. MBFE assumes a jointly-Gaussian distribution between clean and noisy-speech feature vectors, and computes the posterior of the clean one from the observed noisy one. Given the mean and covariance of the clean-speech feature vector, and a non-linear observation model relating the clean and noisy-speech feature vectors, MBFE estimates the parameters of the joint distribution using the first-order vector Taylor series (VTS) [2]. However, as shown in [3], the error incurred by the linearization leads to a degraded performance of the feature enhancer.

The problem of parameter estimation involving a non-linear observation model has been studied in the context of acoustic model adaptation as well. To approximately solve this non-linear estimation problem, techniques like log-normal approximation [4], Monte Carlo method (data-driven parallel model combination) [4], statistical linear approximation [5], as well as VTS [2, 6, 7], have been used. However, the approximation error is not negligible, and damages the performance of speech recognizers.

Recently, a novel technique for nonlinear filtering, named *unscented filtering (UF)* [8], is proposed in the automatic control community. The technique is getting more and more popular as a better substitute for the extended Kalman filter (EKF), which had been used for tens of years as the most popular choice for nonlinear filtering. The key idea with this new filter is the replacement of Taylor series approximation at the core of EKF with a novel approximation technique called the *unscented transformation (UT)*. The resulting algorithm is significantly more accurate than EKF, while the computational cost is the same order as that of EKF.

The present paper proposes the use of the unscented transformation as a better substitute for VTS in the context of feature enhancement. UT uses a minimal number of samples propagated through a non-linear transformation to calculate the statistics of the transformed distribution. By avoiding the linearization of the observation model, the parameters are calculated more accurately, leading to an improved performance of the feature enhancer. Hu and Huo [9] have made a similar attempt for acoustic model adaptation, and had a significantly better result compared with the VTS-based one. Stouten *et al.* [10] have applied UT for feature enhancement in the form of unscented filtering as a better substitute for EKF, but its application is not limited to filtering problems; it can be used more efficiently and effectively in frame-independent feature enhancement schemes. Also, a detailed comparison with VTS, particularly with a higher-order one, has never been reported before. Experimental results show that UT is significantly more accurate than VTS, interestingly an opposite conclusion to the UF-vs-EKF comparison reported in [10]. With the introduction of the state-of-the-art approximation technique, a new feature enhancement algorithm which is powerful and fast is realized.

2. FEATURE ENHANCEMENT

In this section, we briefly review the joint-distribution-based feature enhancement framework, which has been used in MBFE [1] and others [11, 3]. Let $\mathbf{x}$ and $\mathbf{y}$ denote clean-speech and noisy-speech feature vectors of a frame, respectively. The quantities that we want to estimate are the posterior mean and covariance of $\mathbf{x}$ given $\mathbf{y}$, respectively denoted by $\boldsymbol{\mu}_{x|y}$

and $\Sigma_{x|y}$. The posterior covariance is an important quantity to be estimated, since the decoding performance could be improved largely by exploiting the feature enhancement *uncertainty* (i.e. posterior covariance) in the *uncertainty decoding* framework [12]. Suppose that the joint distribution of $\mathbf{x}$ and $\mathbf{y}$ follows a mixture of distributions,

$$p(\mathbf{x},\mathbf{y}) = \sum_{k=1}^{M} \pi_k p(\mathbf{x},\mathbf{y}|k), \tag{1}$$

where M is the number of distributions, and π_k is the mixture weight of k-th distribution. Each distribution is assumed to be a Gaussian as

$$p(\mathbf{x},\mathbf{y}|k) = \mathcal{N}\left(\begin{bmatrix}\mathbf{x}\\ \mathbf{y}\end{bmatrix}; \begin{bmatrix}\boldsymbol{\mu}_x^{(k)}\\ \boldsymbol{\mu}_y^{(k)}\end{bmatrix}, \begin{bmatrix}\Sigma_x^{(k)} & \Sigma_{xy}^{(k)}\\ \Sigma_{yx}^{(k)} & \Sigma_y^{(k)}\end{bmatrix}\right), \tag{2}$$

where $\mathcal{N}(\cdot;\boldsymbol{\mu},\Sigma)$ is a Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance Σ. A set of parameters, π_k, $\boldsymbol{\mu}_x^{(k)}$, $\Sigma_x^{(k)}$, $\boldsymbol{\mu}_y^{(k)}$, $\Sigma_y^{(k)}$, $\Sigma_{yx}^{(k)}$, are assumed to be known. Then, the posterior distribution of $\mathbf{x}$ given $\mathbf{y}$ becomes a mixture of conditional Gaussians as

$$p(\mathbf{x}|\mathbf{y}) = \sum_{k=1}^{M} p(k|\mathbf{y})\mathcal{N}(\mathbf{x};\boldsymbol{\mu}_{x|y}^{(k)},\Sigma_{x|y}^{(k)}), \tag{3}$$

where

$$p(k|\mathbf{y}) = \frac{\pi_k \mathcal{N}(\mathbf{y};\boldsymbol{\mu}_y^{(k)};\Sigma_y^{(k)})}{\sum_{k'}\pi_{k'}\mathcal{N}(\mathbf{y};\boldsymbol{\mu}_y^{(k')};\Sigma_y^{(k')})}, \tag{4}$$

$$\boldsymbol{\mu}_{x|y}^{(k)} = \boldsymbol{\mu}_x^{(k)} + \Sigma_{xy}^{(k)}(\Sigma_y^{(k)})^{-1}(\mathbf{y}-\boldsymbol{\mu}_y^{(k)}), \tag{5}$$

$$\Sigma_{x|y}^{(k)} = \Sigma_x^{(k)} - \Sigma_{xy}^{(k)}(\Sigma_y^{(k)})^{-1}\Sigma_{yx}^{(k)}. \tag{6}$$

Finally, the posterior mean and covariance of $\mathbf{x}$ given $\mathbf{y}$ are calculated from the mixture of conditional Gaussians as

$$\boldsymbol{\mu}_{x|y} = \sum_k p(k|\mathbf{y})\boldsymbol{\mu}_{x|y}^{(k)}, \tag{7}$$

$$\Sigma_{x|y} = \sum_k p(k|\mathbf{y})\left\{\Sigma_{x|y}^{(k)} + (\boldsymbol{\mu}_{x|y}^{(k)}-\boldsymbol{\mu}_{x|y})(\boldsymbol{\mu}_{x|y}^{(k)}-\boldsymbol{\mu}_{x|y})^\top\right\}. \tag{8}$$

3. PARAMETER ESTIMATION

In the joint-distribution-based feature enhancement presented in section 2, parameters, $\boldsymbol{\mu}_y^{(k)}$, $\Sigma_y^{(k)}$, $\Sigma_{yx}^{(k)}$, are required to compute (4), (5) and (6). Although these parameters could be estimated from stereo training samples of $\mathbf{x}$ and $\mathbf{y}$, as was done in [11], a feature enhancer trained with a specific set of noisy data generally performs poorly in *unknown* noise environments. It is preferable, therefore, to estimate these parameters *on-the-fly*. The resulting feature enhancer can adjust itself to a changing noise environment, and does not limit itself to work in a predefined set of environments.

Let $\mathbf{x}$, $\mathbf{n}$, and $\mathbf{y}$ denote clean-speech, noise, and noisy-speech feature vectors, respectively. The means and covariances of $\mathbf{x}$ and $\mathbf{n}$, denoted by $\boldsymbol{\mu}_x^{(k)}$, $\Sigma_x^{(k)}$, $\boldsymbol{\mu}_n$, Σ_n, are assumed to be known (hereafter, the superscript k is omitted for brevity). We also assume that the observation model relating $\mathbf{x}$ and $\mathbf{n}$ to $\mathbf{y}$ is known. If the Mel-frequency cepstral coefficients are used as features, it becomes [13]

$$\mathbf{y} = \mathbf{f}(\mathbf{x},\mathbf{n}) = C\log\left(\exp\left(C^{-1}\mathbf{x}\right) + \exp\left(C^{-1}\mathbf{n}\right)\right), \tag{9}$$

where C is the DCT matrix, and its inverse represents the inverse-DCT. Given these conditions, the problem is to estimate the mean and covariance of $\mathbf{y}$, denoted by $\boldsymbol{\mu}_y$ and Σ_y, and the cross-covariance between $\mathbf{y}$ and $\mathbf{x}$, denoted by Σ_{yx}. The problem involves non-linearity, and no analytical solution exists. In the following, a conventional approach using the truncated vector Taylor series is firstly reviewed. Then, our approach using the unscented transformation is described.

3.1. Vector Taylor Series

For notational convenience, let $\mathbf{z}$ denote $[\mathbf{x}^\top\mathbf{n}^\top]^\top$, with its mean and covariance

$$\boldsymbol{\mu}_z = \begin{bmatrix}\boldsymbol{\mu}_x\\ \boldsymbol{\mu}_n\end{bmatrix}, \quad \Sigma_z = \begin{bmatrix}\Sigma_x & O\\ O & \Sigma_n\end{bmatrix}. \tag{10}$$

Using the vector Taylor series expansion of $\mathbf{f}(\cdot)$ around $\boldsymbol{\mu}_z$, statistics related to $\mathbf{y}$ are calculated as [8]

$$\boldsymbol{\mu}_y = \mathbf{f}(\boldsymbol{\mu}_z) + \mathbb{E}\left[\frac{D_{\Delta z}^2\mathbf{f}}{2}\right] + \ldots, \tag{11}$$

$$\Sigma_y = \mathbb{E}\left[D_{\Delta z}\mathbf{f}(D_{\Delta z}\mathbf{f})^\top\right] + \mathbb{E}\left[\frac{D_{\Delta z}^2\mathbf{f}(D_{\Delta z}^2\mathbf{f})^\top}{4}\right] - \mathbb{E}\left[\frac{D_{\Delta z}^2\mathbf{f}}{2}\right]\mathbb{E}\left[\frac{D_{\Delta z}^2\mathbf{f}}{2}\right]^\top + \ldots, \tag{12}$$

$$\Sigma_{yz} = \begin{bmatrix}\Sigma_{yx} & \Sigma_{yn}\end{bmatrix} = \mathbb{E}\left[D_{\Delta z}\mathbf{f}(\boldsymbol{\Delta}\mathbf{z})^\top\right] + \ldots, \tag{13}$$

where

$$D_{\Delta z}^i\mathbf{f} = \left(\sum_j \Delta z_j \frac{\partial}{\partial z_j}\right)^i \mathbf{f}(\mathbf{z})\Big|_{\mathbf{z}=\boldsymbol{\mu}_z}, \tag{14}$$

and $\boldsymbol{\Delta}\mathbf{z}$ is a Gaussian random vector with mean $\mathbf{0}$ and covariance Σ_z. Third and higher-order terms are omitted. Note that moments of arbitrary-order can be calculated for a Gaussian random vector if Σ_z is given, and the required expectations in above equations can be calculated easily.

In MBFE [1], the first-order VTS was used, i.e. $\mathbf{f}(\cdot)$ was linearly approximated, and the second and higher-order terms in (11), (12), and (13) were truncated. However, $\mathbf{f}(\cdot)$ is highly non-linear, and the linear approximation of it leads to a severe degradation of the ASR performance [3]. To reduce the approximation error, a higher-oder VTS could be used, as was recently proposed by Du and Huo [3]. However, the computational cost increases rapidly with increasing the order of VTS, and the derivation and implementation is getting more and more difficult. To overcome the problem of VTS, we propose to use UT for the parameter estimation.

3.2. Unscented Transformation

The basic idea of the unscented transformation comes from the intuition that *it is easier to approximate a probability distribution than it is to approximate an arbitrary nonlinear function or transformation* [8]. Based on this idea, UT approximates a distribution with a set of samples instead of linearly approximates a function with a truncated Taylor series. UT is similar to Monte Carlo (MC) methods in its spirit, but different in that UT uses a minimal number of samples, called *sigma points*, which are carefully chosen deterministically, while MC uses a large number of randomly chosen

samples, leading to a prohibitively high computational cost (e.g. data-driven parallel model combination [4]).

Let us use $\mathbf{z}$ defined in the previous section, and N_z denote its dimension. A set of sigma points, $\{\mathbf{z}^{(i)}\}_{i=0}^{p}$, where p equals $2N_z$, are generated as [8]

$$\mathbf{z}^{(i)} = \boldsymbol{\mu}_z \qquad (i=0), \tag{15}$$

$$\mathbf{z}^{(i)} = \boldsymbol{\mu}_z + \left(\sqrt{\frac{N_z}{1-W^{(0)}}\Sigma_z}\right)_i \quad (i=1\ldots N_z), \tag{16}$$

$$\mathbf{z}^{(i)} = \boldsymbol{\mu}_z - \left(\sqrt{\frac{N_z}{1-W^{(0)}}\Sigma_z}\right)_{i-N_z} \quad (i=N_z+1\ldots 2N_z). \tag{17}$$

Their associated weights, $\{W^{(i)}\}_{i=0}^{p}$, are defined as $W^{(i)} = (1-W^{(0)})/(2N_z)$ for $i=1,\ldots,2N_z$, where $W^{(0)}$ is a parameter of UT, and is set in this work as $W^{(0)} = 1 - N_z/3$ to match some of the fourth-order moments (kurtoses) to those of a Gaussian. Note that $\left(\sqrt{c\Sigma_z}\right)_i$ represents i-th column (or row) vector of the matrix square root of $c\Sigma_z$ (matrix Σ_z multiplied by a scalar c). For each of the sigma points $\mathbf{z}^{(i)}$, the corresponding $\mathbf{y}^{(i)}$ is generated using the non-linear function, $\mathbf{y} = \mathbf{f}(\mathbf{x},\mathbf{n})$. After generating a set of sigma points, statistics related to $\mathbf{y}$ are finally calculated using those weighted samples as

$$\boldsymbol{\mu}_y = \sum_{i=0}^{p} W^{(i)}\mathbf{y}^{(i)}, \tag{18}$$

$$\Sigma_y = \sum_{i=0}^{p} W^{(i)}(\mathbf{y}^{(i)}-\boldsymbol{\mu}_y)(\mathbf{y}^{(i)}-\boldsymbol{\mu}_y)^\top, \tag{19}$$

$$\Sigma_{yx} = \sum_{i=0}^{p} W^{(i)}(\mathbf{y}^{(i)}-\boldsymbol{\mu}_y)(\mathbf{x}^{(i)}-\boldsymbol{\mu}_x)^\top. \tag{20}$$

It is noted that many other sigma-point selection schemes could be used instead of the one described here.

Figure 1 depicts the mechanism of UT, and compares the performance of UT against VTS. As a simplified observation model, $y = \log(e^z+1)$ was used with $\mu_z = -2$, $\sigma_z = 5$. UT propagates sigma points (circles) through the non-linear function, and calculates μ_y and σ_y using those weighted samples, whereas VTS linearizes the function, and linearly transforms the mean and variance. Compared with VTS, UT's estimates of μ_y and σ_y were much closer to the *true* values, which were calculated via the Monte Carlo (MC) method. In this example, the number of sigma points is three, because $N_z = 1$, which is much smaller than the number required for MC. UT carefully selects a minimal number of samples to realize a computationally efficient algorithm.

UT has a number of advantages over VTS. First of all, it can estimate $\boldsymbol{\mu}_y$, Σ_y, and Σ_{yx} more accurately. A theoretical analysis has shown that UT calculates the mean and covariance correctly to the third-order of Taylor series for any non-linear function if the input is Gaussian (second if non-Gaussian) [8]. Secondly, cumbersome derivation of partial derivatives of $\mathbf{f}(\cdot)$ is not needed. We only need to know the form of $\mathbf{f}(\cdot)$, and even a non-differentiable one could be used. Thirdly, the computational cost of UT is roughly the same as (or modestly higher than) that of VTS. In summary, UT is more accurate, easier to implement, and comparably fast as VTS.

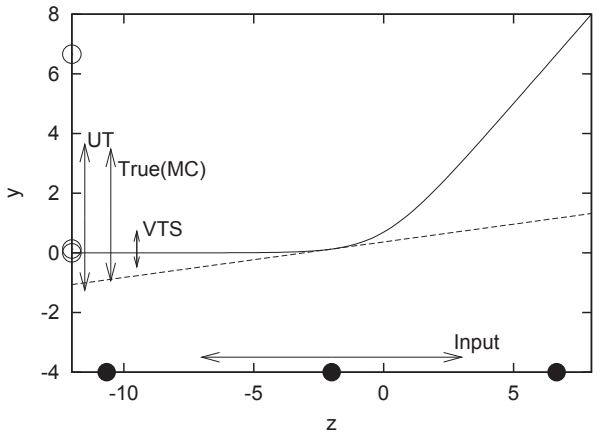

Fig. 1. A comparison of UT and VTS on a simplified observation model. UT propagates sigma points (circles) through the non-linear function (solid) to calculate μ_y and σ_y. VTS linearly approximates the function (dashed). Each vertical arrow indicates the range $[\mu_y - \sigma_y, \mu_y + \sigma_y]$.

4. EXPERIMENTS

4.1. Experimental Setting

The proposed method was evaluated on Aurora2 [14], a standard benchmark set consisting of noisy digit strings in English. Various types of noises were artificially added to clean utterances with SNR ranging from 0 to 20dB. In addition, utterances of Set C were filtered to simulate a channel mismatched condition. The Mel-frequency cepstral coefficients of c_0 to c_{12} were used as features, where power spectrum is used instead of magnitude one. After feature enhancement, Δ and $\Delta\Delta$ features were added. The acoustic model was trained with clean utterances, where each digit was modeled by a 16-state HMM with three Gaussians per state. A Gaussian mixture model with 256 components representing the prior distribution of $\mathbf{x}$ was trained with the same utterances. The first 20 and last 20 frames of each utterance were used to calculate the mean and covariance of noise. The Viterbi decoder of HTK [15] was modified to exploit the uncertainty $\Sigma_{x|y}$ in the uncertainty decoding framework, where the uncertainty of Δ and $\Delta\Delta$ features were calculated from that of static features in the same manner as in [12]. Diagonal covariances were used for $\Sigma_x^{(k)}$, Σ_n, and $\Sigma_{x|y}$.

4.2. Performance

Table 1 shows the comparative performance of the feature enhancers with different parameter estimation techniques. Compared with the baseline which did not use any feature enhancement, all of the feature enhancers reduced the word error rate by 65% or more relatively. The second-order VTS performed better than the first-order one, indicating the importance of higher-order (non-linear) terms of the series. The unscented transformation performed even better than the second-order VTS, leading us to believe that the non-linearity incurred by the non-linear observation model (9) can be handled more accurately with UT than with VTS. By exploiting the uncertainty of feature enhancement, a clear improvement was observed in all conditions.

Table 1. Word accuracy (%) on Aurora2. The acoustic model was trained with clean data. 'UD' indicates uncertainty decoding.

	UD	A	B	C	Ave
Baseline	-	59.10	55.50	66.49	59.14
First-order VTS	-	86.90	87.35	82.84	86.27
	yes	87.11	87.92	83.89	86.79
Second-order VTS	-	87.63	87.75	84.04	86.96
	yes	87.83	88.47	84.75	87.47
Unscented	-	87.89	87.89	84.42	87.20
	yes	88.30	88.78	85.41	87.91

4.3. Cost

Table 2 compares the computational cost. The parameters of enhancement, i.e. $\boldsymbol{\mu}_y^{(k)}$, $\Sigma_y^{(k)}$, $\Sigma_{yx}^{(k)}$, were assumed to be constant over an utterance, and computed only once per utterance. In our research C code, UT was about two times more expensive than the first-order VTS in initializing parameters, while at least two times faster than the second-order one. Overall MIPS modestly increased from 117 to 151. The actual computational time measured on a 3.8 GHz machine for enhancing a three-second utterance (=300 frames) was 0.21 second, i.e. 14 times faster than the real time. The computational complexity of VTS grows rapidly as the order gets higher, implying the difficulty of third or higher-order ones regarding the computational cost. It is noted that the use of other sigma-point selection schemes could boost the speed of UT without significantly damaging the performance [8].

Table 2. The computational cost: 'init.' is the # of instructions for computing $\boldsymbol{\mu}_y^{(k)}$, $\Sigma_y^{(k)}$, and $\Sigma_{yx}^{(k)}$; 'per frame' for computing $\boldsymbol{\mu}_{x|y}$ and $\Sigma_{x|y}$; 'MIPS' is the cost (million instructions) per second in enhancing a three-second utterance.

	init.	per frame	MIPS
First-order VTS	93M	0.86M	117
Second-order VTS	477M		245
Unscented	195M		151

4.4. Kullback-Leibler Divergence

Table 3 shows the Kullback-Leibler divergence from the "true" Gaussian distribution to the estimated one, where the "true" one was obtained via the Monte Carlo method with 10,000 samples. The divergence was calculated for $\mathcal{N}(\boldsymbol{\mu}_y, \Sigma_y)$ with ignoring the off-diagonal elements of Σ_y, and was averaged over 256 components and 50 utterances randomly selected from Subway 10dB of Set A. UT was about eight times more accurate than the first-order VTS, and was comparably accurate with the Monte Carlo method with 1,000 samples. Note that UT used only 53 samples, i.e. about 20 times more efficient than the Monte Carlo method. Similar trends were observed with other noises and SNRs.

Table 3. Average Kullback-Leibler divergence from the "true" distribution to the estimated one.

VTS	**Uns.**	Monte Carlo				
		30	100	300	1,000	3,000
.124	.015	.523	.143	.046	.014	.006

5. CONCLUSIONS

The present work improved the Model-Based Feature Enhancement [1] by using the unscented transformation (UT) in place of VTS for parameter estimation. Previously, parameters of the joint distribution of clean and noisy-speech features are calculated by linearly approximating the non-linear observation model with VTS. In the proposed method, the parameters are calculated via UT using a minimal number of samples propagated through the non-linear observation model. By avoiding the linearization, UT calculates the parameters more accurately, leading to an improved performance of the feature enhancer. In the Aurora2 evaluation, UT achieved a relative word error rate reduction of 8.48% compared with VTS, while the computational cost was just modestly higher than that of VTS.

It is noted that UT is easier to implement than VTS. Since the derivation of Jacobians is not needed, development and test of new systems using different observation models can be carried out rapidly. We believe that UT can be applied to a number of non-linear estimation problems which appear in the field of speech processing.

6. REFERENCES

[1] V. Stouten, H. Van hamme, and P. Wambacq, "Model-based feature enhancement with uncertainty decoding for noise robust ASR," *Speech Communication*, 48:1502–1514, 2006.

[2] P. Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, 1996.

[3] 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," in *Proc. Interspeech*, 2008.

[4] M.J.F. Gales, *Model-based techniques for noise robust speech recognition*, Ph.D. thesis, University of Cambridge, 1995.

[5] N. S. Kim, "Statistical linear approximation for environment compensation," *Signal Processing Letters*, 5(1):8–10, 1998.

[6] H. Xu, L. Rigazio, and D. Kryze, "Vector Taylor series based joint uncertainty decoding," in *Proc. Interspeech*, 2006.

[7] H. Liao, *Uncertainty Decoding for Noise Robust Speech Recognition*, Ph.D. thesis, University of Cambridge, 2007.

[8] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[9] Y. Hu and Q. Huo, "An HMM compensation approach using unscented transformation for noisy speech recognition," in *Proc. ISCSLP*, 2006.

[10] V. Stouten, H. Van hamme, and P. Wambacq, "Kalman and Unscented Kalman Filter Feature Enhancement for Noise Robust ASR," in *Proc. Interspeech*, 2005.

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

[12] L. Deng, J. Droppo, and A. Acero, "Dynamic compensation of HMM variances using the feature enhancement uncertainty computed from a parametric model of speech distortion," *IEEE Trans. SAP*, 13(3):412–421, 2005.

[13] A. Acero, *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Ph.D. thesis, Carnegie Mellon University, 1990.

[14] D. Pearce and H.-G. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ICSLP*, 2000.

[15] S. Young *et al.*, *The HTK Book*, Cambridge University Engineering Department, 2006.