A Feature Compensation Approach Using High-Order Vector Taylor Series Approximation of an Explicit Distortion Model for Noisy Speech Recognition

Jun Du and Qiang Huo, *Member, IEEE*

***Abstract*—This paper presents a new feature compensation approach to noisy speech recognition by using high-order vector Taylor series (HOVTS) approximation of an explicit model of environmental distortions. Formulations for maximum-likelihood (ML) estimation of both additive noises and convolutional distortions, and minimum mean squared error (MMSE) estimation of clean speech are derived. Experimental results on Aurora2 and Aurora4 benchmark databases, where the modeling assumption of the distortion model is more accurate, demonstrate that the standard HOVTS-based feature compensation approaches achieve consistently significant improvement in recognition accuracy compared to traditional standard first-order VTS-based approach. For a real-world in-vehicle connected digits recognition task on Aurora3 benchmark database where the modeling assumption of the distortion model is less accurate, modifications are necessary to make VTS-based feature compensation approaches work. In this case, the second-order VTS-based approach performs only slightly better than the first-order VTS-based approach.**

***Index Terms*—Distortion model, feature compensation, noise robustness, robust speech recognition, vector Taylor series (VTS).**

I. Introduction

MOST of current automatic speech recognition (ASR) systems use Mel-frequency cepstral coefficients (MFCCs) and their derivatives as speech features, and a set of Gaussian mixture continuous density HMMs (CDHMMs) for modeling basic speech units. It is well known that the performance of such an ASR system trained with clean speech will degrade significantly when the testing speech is corrupted by additive noises and convolutional distortions. One type of approaches to dealing with the above problem is the so-called feature compensation approach using *explicit* model of environmental distortions (e.g., [1]), which is also the topic of this paper.

For our approach, it is assumed that in time domain, "corrupted" speech $y[t]$ is subject to the following *explicit* distortion model:

$$y[t] = x[t] \circledast h[t] + n[t] \tag{1}$$

where independent signals $x[t]$, $h[t]$ and $n[t]$ represent the tth sample of clean speech, the convolutional (e.g., transducer and transmission channel) distortion and the additive noise, respectively. In log-power-spectral domain, the distortion model can be expressed *approximately* (e.g., [1]) as

$$\exp(\mathbf{y}^{\mathrm{l}}) = \exp(\mathbf{x}^{\mathrm{l}} + \mathbf{h}^{\mathrm{l}}) + \exp(\mathbf{n}^{\mathrm{l}}) \tag{2}$$

where $\mathbf{y}^{\mathrm{l}}$, $\mathbf{x}^{\mathrm{l}}$, $\mathbf{h}^{\mathrm{l}}$, and $\mathbf{n}^{\mathrm{l}}$ are log power-spectra of noisy speech, clean speech, convolutional term and noise, respectively. In MFCC domain, the distortion model becomes

$$\mathbf{y}^{\mathrm{c}} = \mathbf{C} \log[\exp(\mathbf{C}^{+}(\mathbf{x}^{\mathrm{c}} + \mathbf{h}^{\mathrm{c}})) + \exp(\mathbf{C}^{+}\mathbf{n}^{\mathrm{c}})] \tag{3}$$

where $\mathbf{C}$ is a $D^{\mathrm{c}} \times D^{\mathrm{l}}$ truncated discrete cosine transform (DCT) matrix, $\mathbf{C}^{+}$ denotes the Moore–Penrose inverse of $\mathbf{C}$ (refer to [12] for details), D^{c} is the dimension of MFCC feature vector, and D^{l} is the number of channels of the Mel-frequency filterbank used in MFCC feature extraction. In most of the current ASR systems, $D^{\mathrm{c}} < D^{\mathrm{l}}$. The log and exp functions in the above equations operate element-by-element on the corresponding vectors. The nonlinear nature of the above distortion model makes statistical modeling and inference of the above variables difficult; therefore, certain approximations have to be made.

Understandably, a simple linear approximation, namely the first-order vector Taylor series (VTS) approximation, has been tried in the past (e.g., [14], [16], and [17]). There are also efforts in using high-order VTS (HOVTS) to improve the above first-order VTS approximation. In [13], the nonlinear distortion function for additive noise only is first expanded using HOVTS. Then a linear function is found to approximate the above HOVTS by minimizing the mean-squared error incurred by this approximation. Given the linear function, the remaining inference is the same as in using the traditional first-order VTS to approximate the nonlinear distortion function directly. In [6], HOVTS is used to approximate the nonlinear portion of the distortion function by expanding with respect to $\mathbf{n}^{\mathrm{l}} - \mathbf{x}^{\mathrm{l}}$ instead of $(\mathbf{x}^{\mathrm{l}}, \mathbf{n}^{\mathrm{l}})$. Both approaches work for each feature dimension independently by ignoring correlations between different channels of the filterbank. In [21], the above nonlinear distortion function is approximated by a second-order VTS. Using this relation, the mean vector of the relevant noisy speech feature vector can be derived, which naturally includes a term related to the second-order term in HOVTS. In our previous work [7]–[9], we extended the above work in the following ways: 1) the nonlinear distortion function for both additive noise and convolutional distortion can be approximated by HOVTS with any order; 2) the

Manuscript received August 02, 2010; revised December 28, 2010; accepted March 01, 2011. Date of publication March 17, 2011; date of current version August 19, 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Brian Mak.

Dr. J. Du is with the Visual Computing Group, Microsoft Research Asia (MSRA), Beijing 100190, China (e-mail: jundu@microsoft.com).

Dr. Q. Huo is with Speech Group, Microsoft Research Asia (MSRA), Beijing 100190, China (e-mail: qianghuo@microsoft.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASL.2011.2129508

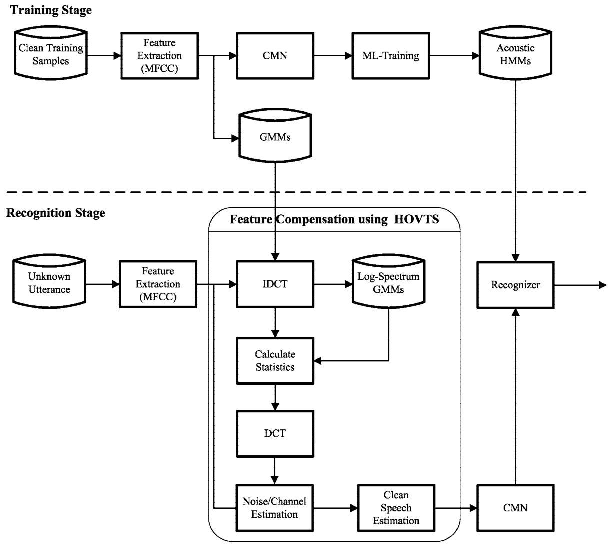

Fig. 1. Flowchart of our feature compensation approach.

required sufficient statistics are derived for estimating model parameters of additive noise and convolutional distortion, and clean speech feature vector; and 3) correlations between different channels of the filterbank can be considered. In this paper, we expand on our previous work, providing a more detailed description of the formulation and derivation of the proposed approach, additional implementation details, new experiments, and an expanded discussion of the results.

The rest of the paper is organized as follows. In Section II, we give an overview of the general formulation of our feature compensation approach. In Section III, we present the detailed formulation of how to calculate the required sufficient statistics based on HOVTS approximation. In Section IV, some implementation issues are discussed. In Section V, we report experimental results and finally we conclude the paper in Section VI. To make the paper more accessible, in Appendix A, we summarize how to derive a procedure for the maximum-likelihood (ML) estimation of both additive noise and convolutional distortion.

II. Feature Compensation Approach

The flowchart of our feature compensation approach is illustrated in Fig. 1. In the training stage, a Gaussian mixture model (GMM)

$$p(\mathbf{x}^{\rm c}_t)=\sum_{m=1}^{M}\omega_m\mathcal{N}(\mathbf{x}^{\rm c}_t;\boldsymbol{\mu}^{\rm c}_{\mathbf{x},m},\boldsymbol{\Sigma}^{\rm c}_{\mathbf{x},m})$$

is trained from clean speech using MFCC features without cepstral mean normalization (CMN), where $\boldsymbol{\mu}^{\rm c}_{\mathbf{x},m}$, $\boldsymbol{\Sigma}^{\rm c}_{\mathbf{x},m}$, and ω_m are mean vector, diagonal covariance matrix, and mixture weight of the mth Gaussian component, respectively. Let us assume that for each sentence, the noise feature vector $\mathbf{n}^{\rm c}$ in cepstral domain follows a Gaussian probability density function (pdf) with a mean vector $\boldsymbol{\mu}^{\rm c}_{\mathbf{n}}$ and a diagonal covariance matrix $\boldsymbol{\Sigma}^{\rm c}_{\mathbf{n}}$. Let us further assume that the term $\mathbf{h}^{\rm c}$ corresponding to convolutional distortion has a pdf of the Kronecker delta function $\delta(\mathbf{h}^{\rm c}-\mathbf{h}^{\rm c}_{\rm const})$, where $\mathbf{h}^{\rm c}_{\rm const}$ is an unknown deterministic vector.

In the recognition stage, the above unknown distortion model parameters can be estimated as follows.

Step 1) Initialization: We first estimate the initial noise model parameters in cepstral domain by taking the sample mean and covariance of the MFCC features from the first several (ten in our experiments) frames of the unknown utterance, and set $\mathbf{h}^{\rm c}_{\rm const}$ as a zero vector.

Step 2) Define a new random vector, $\mathbf{z}^{\rm c}=\mathbf{x}^{\rm c}+\mathbf{h}^{\rm c}$, whose pdf can be derived as follows:

$$p(\mathbf{z}^{\rm c}_t)=\sum_{m=1}^{M}\omega_m\mathcal{N}(\mathbf{z}^{\rm c}_t;\boldsymbol{\mu}^{\rm c}_{\mathbf{x},m}+\mathbf{h}^{\rm c}_{\rm const},\boldsymbol{\Sigma}^{\rm c}_{\mathbf{x},m}). \tag{4}$$

Then transform all parameters from cepstral domain to log-power-spectral domain as follows:

$$\boldsymbol{\mu}^{\rm l}_{\mathbf{z},m}=\mathbf{C}^{+}(\boldsymbol{\mu}^{\rm c}_{\mathbf{x},m}+\mathbf{h}^{\rm c}_{\rm const}) \tag{5}$$

$$\boldsymbol{\Sigma}^{\rm l}_{\mathbf{z},m}=\mathbf{C}^{+}\boldsymbol{\Sigma}^{\rm c}_{\mathbf{x},m}(\mathbf{C}^{+})^{\top} \tag{6}$$

$$\boldsymbol{\mu}^{\rm l}_{\mathbf{n}}=\mathbf{C}^{+}\boldsymbol{\mu}^{\rm c}_{\mathbf{n}} \tag{7}$$

$$\boldsymbol{\Sigma}^{\rm l}_{\mathbf{n}}=\mathbf{C}^{+}\boldsymbol{\Sigma}^{\rm c}_{\mathbf{n}}(\mathbf{C}^{+})^{\top} \tag{8}$$

where the superscripts "l" and "c" indicate the log-power-spectral domain and cepstral domain, respectively.

Step 3) In log-power-spectral domain, use HOVTS approximation to calculate the relevant statistics, $\boldsymbol{\mu}^{\rm l}_{\mathbf{y},m}$, $\boldsymbol{\Sigma}^{\rm l}_{\mathbf{y},m}$, $\boldsymbol{\Sigma}^{\rm l}_{\mathbf{zy},m}$, $\boldsymbol{\Sigma}^{\rm l}_{\mathbf{ny},m}$, which are required for re-estimation of distortion model parameters and estimation of clean speech. The details are given in the next section.

Step 4) Transform the above statistics back to cepstral domain as follows:

$$\boldsymbol{\mu}^{\rm c}_{\mathbf{y},m}=\mathbf{C}\boldsymbol{\mu}^{\rm l}_{\mathbf{y},m} \tag{9}$$

$$\boldsymbol{\Sigma}^{\rm c}_{\mathbf{y},m}=\mathbf{C}\boldsymbol{\Sigma}^{\rm l}_{\mathbf{y},m}(\mathbf{C})^{\top} \tag{10}$$

$$\boldsymbol{\Sigma}^{\rm c}_{\mathbf{zy},m}=\mathbf{C}\boldsymbol{\Sigma}^{\rm l}_{\mathbf{zy},m}(\mathbf{C})^{\top} \tag{11}$$

$$\boldsymbol{\Sigma}^{\rm c}_{\mathbf{ny},m}=\mathbf{C}\boldsymbol{\Sigma}^{\rm l}_{\mathbf{ny},m}(\mathbf{C})^{\top}\,. \tag{12}$$

Step 5) Use the following updating formulas (to be derived in Appendix A) to re-estimate the distortion model parameters:

$$\bar{\boldsymbol{\mu}}_{\mathbf{n}}=\frac{\sum\limits_{t=1}^{T}\sum\limits_{m=1}^{M}P(m|\mathbf{y}_t)E_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t,m]}{\sum\limits_{t=1}^{T}\sum\limits_{m=1}^{M}P(m|\mathbf{y}_t)} \tag{13}$$

$$\bar{\boldsymbol{\Sigma}}_{\mathbf{n}}=\frac{\sum\limits_{t=1}^{T}\sum\limits_{m=1}^{M}P(m|\mathbf{y}_t)E_{\mathbf{n}}[\mathbf{n}_t\mathbf{n}_t^{\top}|\mathbf{y}_t,m]}{\sum\limits_{t=1}^{T}\sum\limits_{m=1}^{M}P(m|\mathbf{y}_t)}-\bar{\boldsymbol{\mu}}_{\mathbf{n}}\bar{\boldsymbol{\mu}}_{\mathbf{n}}^{\top} \tag{14}$$

$$\begin{aligned}\bar{\mathbf{h}}_{\rm const}=&\left[\sum_{t=1}^{T}\sum_{m=1}^{M}P(m|\mathbf{y}_t)\boldsymbol{\Sigma}^{-1}_{\mathbf{x},m}\right]^{-1}\\&\left[\sum_{t=1}^{T}\sum_{m=1}^{M}P(m|\mathbf{y}_t)\boldsymbol{\Sigma}^{-1}_{\mathbf{x},m}(E_{\mathbf{z}}[\mathbf{z}_t|\mathbf{y}_t,m]-\boldsymbol{\mu}_{\mathbf{x},m})\right]\end{aligned} \tag{15}$$

where

$$P(m|\mathbf{y}_t) = \frac{\omega_m p_{\mathbf{y}}(\mathbf{y}_t|m)}{\sum\limits_{l=1}^{M} \omega_l p_{\mathbf{y}}(\mathbf{y}_t|l)} . \tag{16}$$

In the above equations, we have dropped the cepstral domain indicator "c" in relevant variables for notational convenience. Furthermore, $p_{\mathbf{y}}(\mathbf{y}_t) = \sum_{m=1}^{M} \omega_m p_{\mathbf{y}}(\mathbf{y}_t|m)$, is the pdf of the noisy speech $\mathbf{y}_t$, where the true $p_{\mathbf{y}}(\mathbf{y}_t|m)$ is approximated by a Gaussian pdf, $\mathcal{N}(\mathbf{y}_t; \boldsymbol{\mu}_{\mathbf{y},m}, \boldsymbol{\Sigma}_{\mathbf{y},m})$, via "moment-matching." $E_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t, m]$, $E_{\mathbf{n}}[\mathbf{n}_t\mathbf{n}_t^\top|\mathbf{y}_t, m]$ and $E_{\mathbf{z}}[\mathbf{z}_t|\mathbf{y}_t, m]$ are the relevant conditional expectations evaluated as follows:

$$E_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t, m] = \boldsymbol{\mu}_{\mathbf{n}} + \boldsymbol{\Sigma}_{\mathbf{ny},m}\boldsymbol{\Sigma}_{\mathbf{y},m}^{-1}(\mathbf{y}_t - \boldsymbol{\mu}_{\mathbf{y},m}) \tag{17}$$

$$\begin{aligned} E_{\mathbf{n}}[\mathbf{n}_t\mathbf{n}_t^\top|\mathbf{y}_t, m] &= E_{\mathbf{n}}[\mathbf{n}_t|\mathbf{y}_t, m]E_{\mathbf{n}}^\top[\mathbf{n}_t|\mathbf{y}_t, m] \\ &\quad + \boldsymbol{\Sigma}_{\mathbf{n}} - \boldsymbol{\Sigma}_{\mathbf{ny},m}\boldsymbol{\Sigma}_{\mathbf{y},m}^{-1}\boldsymbol{\Sigma}_{\mathbf{y}n,m} \end{aligned} \tag{18}$$

$$\begin{aligned} E_{\mathbf{z}}[\mathbf{z}_t|\mathbf{y}_t, m] &= (\boldsymbol{\mu}_{\mathbf{x},m} + \mathbf{h}_{\text{const}}) \\ &\quad + \boldsymbol{\Sigma}_{\mathbf{zy},m}\boldsymbol{\Sigma}_{\mathbf{y},m}^{-1}(\mathbf{y}_t - \boldsymbol{\mu}_{\mathbf{y},m}) . \end{aligned} \tag{19}$$

Step 6) Repeat Step 2 to Step 5 N_{VTS} times.

Given the noisy speech and the estimated distortion model parameters, the minimum mean-squared error (MMSE) estimation of clean speech feature vector in cepstral domain can be calculated as

$$\hat{\mathbf{x}}_t = E_{\mathbf{x}}\left[\mathbf{x}_t|\mathbf{y}_t\right] = \sum_{m=1}^{M} P(m|\mathbf{y}_t)E_{\mathbf{x}}\left[\mathbf{x}_t|\mathbf{y}_t, m\right] \tag{20}$$

where $E_{\mathbf{x}}\left[\mathbf{x}_t|\mathbf{y}_t, m\right]$ is the conditional expectation of $\mathbf{x}_t$ given $\mathbf{y}_t$ for the mth mixture component and can be evaluated as follows:

$$\begin{aligned} E_{\mathbf{x}}[\mathbf{x}_t|\mathbf{y}_t, m] &= E_{\mathbf{z}}[\mathbf{z}_t|\mathbf{y}_t, m] - \mathbf{h}_{\text{const}} \\ &= \boldsymbol{\mu}_{\mathbf{x},m} + \boldsymbol{\Sigma}_{\mathbf{zy},m}\boldsymbol{\Sigma}_{\mathbf{y},m}^{-1}(\mathbf{y}_t - \boldsymbol{\mu}_{\mathbf{y},m}) . \end{aligned} \tag{21}$$

The other modules in Fig. 1 are self-explained.

In the next section, we elaborate on how to calculate the required statistics, $\boldsymbol{\mu}_{\mathbf{y},m}^{\mathrm{l}}$, $\boldsymbol{\Sigma}_{\mathbf{y},m}^{\mathrm{l}}$, $\boldsymbol{\Sigma}_{\mathbf{zy},m}^{\mathrm{l}}$, $\boldsymbol{\Sigma}_{\mathbf{ny},m}^{\mathrm{l}}$, using HOVTS approximation of the nonlinear distortion function in (2). For notational convenience, we drop hereinafter the indices related to the frame number, mixture component, and channel index of the filterbank without causing confusions.

III. Computation of Required Statistics

The explicit distortion model in (2) is reformulated in the scalar form as follows:

$$y = f(z, n) = \log(\exp(z) + \exp(n)) \tag{22}$$

where $z = x + h$. Then the K-order Taylor series of $f(z, n)$ with the expansion point (μ_z, μ_n) can be represented as

$$\begin{aligned} f_K(z, n) &= \sum_{k=0}^{K} \frac{1}{k!}\left[(z - \mu_z)\frac{\partial}{\partial z} + (n - \mu_n)\frac{\partial}{\partial n}\right]^k f(\mu_z, \mu_n) \\ &= \sum_{k=0}^{K}\sum_{r=0}^{k} A(k, r)(z - \mu_z)^{k-r}(n - \mu_n)^r \end{aligned} \tag{23}$$

where

$$A(k, r) = \frac{1}{r!(k - r)!}\left.\frac{\partial^k f(z, n)}{\partial z^{k-r}\partial n^r}\right|_{(\mu_z, \mu_n)} \tag{24}$$

and

$$\left.\frac{\partial^k f(z, n)}{\partial z^{k-r}\partial n^r}\right|_{(\mu_z, \mu_n)} = \begin{cases} \log(\exp(\mu_z) + \exp(\mu_n)), & k = 0, r = 0 \\ 1 - \frac{1}{1+\exp(\mu_n-\mu_z)}, & k = 1, r = 1 \\ \frac{1}{1+\exp(\mu_n-\mu_z)}, & k = 1, r = 0 \\ (-1)^{k-r}\sum\limits_{p=1}^{k}\frac{B(k,p)}{[1+\exp(\mu_n-\mu_z)]^p}, & k > 1. \end{cases} \tag{25}$$

When $k > 1$ and $k \geq p \geq 1$, the coefficients $B(k, p)$ in (25) can be evaluated by using the following recursive relation

$$B(k, p) = (p - 1)B(k - 1, p - 1) - pB(k - 1, p) \tag{26}$$

with the initial condition

$$B(1, 1) = -1, B(k, 0) = B(k, k + 1) = 0, \;\; k \geq 1 . \tag{27}$$

For convenience, we also define the following expectations:

$$\begin{aligned} &E_{zn}^{i}[g(z, n)] \\ &= \iint g(z^i, n^i)p_{zn}(z^i, n^i)dz^i dn^I \end{aligned} \tag{28}$$

$$\begin{aligned} &E_{zn}^{ij}[g(z, n), h(z, n)] \\ &= \iiiint g(z^i, n^i)h(z^j, n^j)p_{zn}(z^i, z^j, n^i, n^j)dz^i dz^j dn^i dn^j \end{aligned} \tag{29}$$

where $g(z^i, n^i)$ and $h(z^j, n^j)$ are two general functions, i and j are dimensional indices.

Given the above notations and results, we summarize in the following subsections the main statistics required in implementing our feature compensation approach.

A. Calculating $\mu_y(i)$

Let us use $\mu_y(i)$ to denote the ith element of the vector $\boldsymbol{\mu}_{\mathbf{y}}$. Using the definition of the mean parameter, we have

$$\begin{aligned} \mu_y(i) &\doteq E_{zn}^{i}[f_K(z, n)] \\ &= \sum_{k=0}^{K}\sum_{r=0}^{k} A^i(k, r)E_{zn}^{i}[(z - \mu_z)^{k-r}(n - \mu_n)^r] \\ &= \sum_{k=0}^{K}\sum_{r=0}^{k} A^i(k, r)M_n^i(r)M_z^i(k - r) \end{aligned} \tag{30}$$

where

$$M_{\Delta}^{i}(p) = \begin{cases} 0, & \text{if } p \text{ is odd} \\ (p - 1)!!\sigma_{\Delta}^{p}(i), & \text{otherwise.} \end{cases} \tag{31}$$

Δ represents "z" or "n." $A^i(k, r)$ is the value of (24) for the ith dimension. And for p is even, we have

$$(p - 1)!! = \prod_{l=1}^{p/2}(2l - 1) \tag{32}$$

B. Calculating $\sigma_y^2(i,j)$

Let us use $\sigma_y^2(i,j)$ to denote the (i,j)th element of the matrix $\boldsymbol{\Sigma}_{\mathbf{y}}$. Using the definition of the covariance, we have

$$\begin{aligned}\sigma_y^2(i,j) &\doteq E_{zn}^{ij}[f_K(z,n), f_K(z,n)] - \mu_y(i)\mu_y(j)\\ &= \sum_{k_1=0}^{K}\sum_{r_1=0}^{k_1}\sum_{k_2=0}^{K}\sum_{r_2=0}^{k_2}[A^i(k_1,r_1)A^j(k_2,r_2)\\ &\quad M_n^{ij}(r_1,r_2)M_z^{ij}(k_1-r_1,k_2-r_2)] - \mu_y(i)\mu_y(j)\end{aligned} \tag{33}$$

where

$$M_\Delta^{ij}(p,q) = \begin{cases} 0, & \text{if } p+q \text{ is odd}\\ p!q!2^{-(p+q)/2}\sum\limits_{0\le l\le \min(p,q)}^{p-\mathrm{l}\text{ is even}} & \\ \frac{2^{\mathrm{l}}}{l!(\frac{p-l}{2})!(\frac{q-l}{2})!}\sigma_\Delta^{p-\mathrm{l}}(i,i) & \\ \sigma_\Delta^{2\mathrm{l}}(i,j)\sigma_\Delta^{q-\mathrm{l}}(j,j), & \text{otherwise.}\end{cases} \tag{34}$$

C. Calculating $\sigma_{zy}^2(i,j)$

Let us use $\sigma_{zy}^2(i,j)$ to denote the (i,j)th element of the matrix $\boldsymbol{\Sigma}_{\mathbf{zy}}$. Using the definition of the covariance parameter, we have

$$\begin{aligned}\sigma_{zy}^2(i,j) &= E_{zn}^{ij}[(z-\mu_z),(y-\mu_y)]\\ &= \sum_{k=0}^{K}\sum_{r=0}^{k}A^j(k,r)M_n^j(r)M_z^{ij}(1,k-r)\ .\end{aligned} \tag{35}$$

D. Calculating $\sigma_{ny}^2(i,j)$

Let us use $\sigma_{ny}^2(i,j)$ to denote the (i,j)th element of the matrix $\boldsymbol{\Sigma}_{\mathbf{ny}}$. Using the definition of the covariance parameter, we have

$$\begin{aligned}\sigma_{ny}^2(i,j) &= E_{zn}^{ij}[(n-\mu_n),(y-\mu_y)]\\ &= \sum_{k=0}^{K}\sum_{r=0}^{k}A^j(k,r)M_n^{ij}(1,r)M_z^j(k-r)\ .\end{aligned} \tag{36}$$

IV. Implementation Issues

Although using a higher order VTS to approximate the explicit distortion model in (2) is well-motivated, its effectiveness largely depends on how faithful the adopted distortion model reflects the truth in the unknown utterance to be recognized. As we will demonstrate in the following section on experiments, two heuristic strategies described in the following two subsections may help improve the effectiveness of the proposed feature compensation approach when the distortion model may not be accurate enough.

A. Combined Cepstral Mean Normalization and VTS-Based Feature Compensation Approach

In our experiments of real-world noisy speech recognition on Aurora3 task, we observed that it is helpful to use a combined cepstral mean normalization (CMN) and VTS-based feature compensation approach as illustrated in Fig. 2. In this case, the clean-speech GMM is trained using the CMN-processed MFCC features.

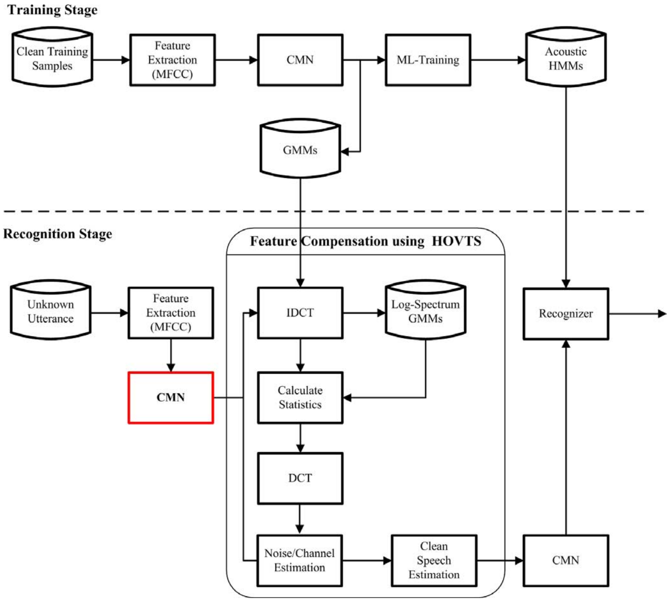

Fig. 2. Flowchart of combined cepstral mean normalization and VTS-based feature compensation approach.

B. Modified Clean Speech Estimation Approaches

In our proposed approach, an important step is to use (20) to estimate clean speech under MMSE criterion. Again, in our experiments on Aurora3 task, we made the following observation: for certain frames $\{\mathbf{y}_t^{\mathrm{c}}\}$ in both high-SNR and low-SNR regions, the term $\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{y},m})^{-1}$ in (21) is very close to zero on quite a few mixture components in (20), which renders $\{\mathbf{y}_t^{\mathrm{c}}\}$ non-differentiable to the mixture components concerned. To deal with this problem, one possible heuristic solution is to constrain the value of $\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{y},m})^{-1}$.

Let us analyze first the calculation of $\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{y},m})^{-1}$ for the first-order VTS case. As we have

$$\begin{aligned}\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{y},m})^{-1} &= \mathbf{C}\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{zy},m}\mathbf{C}^{\top}(\mathbf{C}\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{y},m}\mathbf{C}^{\top})^{-1}\\ &= \mathbf{C}\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{y},m})^{-1}\mathbf{C}^{+}\end{aligned} \tag{37}$$

the value of $\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{c}}_{\mathbf{y},m})^{-1}$ is determined by $\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{y},m})^{-1}$. Furthermore, the main contribution of $\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{y},m})^{-1}$ comes from the diagonal elements of $\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{zy},m}$ and $\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{y},m}$, respectively. If we ignore all the off-diagonal elements of both $\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{zy},m}$ and $\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{y},m}$, then $\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\mathrm{l}}_{\mathbf{y},m})^{-1}$ can be reduced to a diagonal matrix with the ith element defined (the superscript l and the subscript m are ignored for neat notation) as

$$\frac{\sigma_{zy}^2(i)}{\sigma_y^2(i)} = \frac{a(i)\sigma_z^2(i)}{a^2(i)\sigma_z^2(i) + [1-a(i)]^2\sigma_n^2(i)} \tag{38}$$

where

$$a(i) = \frac{1}{1+\exp[n_0(i) - z_0(i)]} = \frac{1}{1+\exp[\mu_n(i) - \mu_z(i)]}\ . \tag{39}$$

In the above equation, $(n_0(i), z_0(i))$ is the expansion point of VTS, which is set to the corresponding mean vectors $(\mu_n(i), \mu_z(i))$. From (38), we can observe that the key term which causes $\boldsymbol{\Sigma}^{\rm c}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\rm c}_{\mathbf{y},m})^{-1}$ close to zero is $a(i)$, which is related to the expansion point as follows:

$$\mu_n(i) \gg \mu_z(i) \Rightarrow a(i) \to 0 \Rightarrow \boldsymbol{\Sigma}^{\rm c}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\rm c}_{\mathbf{y},m})^{-1} \to 0 \quad (40)$$

which could happen under two cases. The first case is that the value of $\mu_n(i)$ is large which happens when the SNR of the current utterance is low. The second case is that the value of $\mu_z(i)$ is small, which can easily happen for a quite few mixture components representing the small "energy" component of distribution space defined in (4).

One way to constrain the value of $\boldsymbol{\Sigma}^{\rm c}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\rm c}_{\mathbf{y},m})^{-1}$ not to be very small, which represents the frame differentiability in (21), is to constrain $a(i)$ which is related with $z_0(i)$ and $n_0(i)$. To achieve this goal, here we adjust the value of $z_0(i)$ and still set $n_0(i)$ to $\mu_n(i)$. This is a natural idea as $\mu_n(i)$ is estimated online using EM algorithm which represents the noise information of the current utterance while $\mu_z(i)$ of some small "energy" components is not suitable to be used as the expansion point $z_0(i)$ due to several assumptions made in the inference. Let us define an SNR measure for the ith dimension as follows:

$$\mathrm{SNR}(i) = 10 \log \frac{\rho^2_{zy}(i)}{1 - \rho^2_{zy}(i)} \quad (41)$$

where

$$\rho_{zy}(i) = \frac{\sigma^2_{zy}(i)}{\sqrt{\sigma^2_z(i)\sigma^2_y(i)}} \quad (42)$$

is the correlation coefficient of z and y. By using (38) to (42), the expansion point $z_0(i)$ can be expressed as

$$z_0(i) = \frac{1}{20}\mathrm{SNR}(i) - \frac{1}{2}\log\frac{\sigma^2_z(i)}{\sigma^2_n(i)} + n_0(i)\,. \quad (43)$$

Consequently, we propose to constrain $z_0(i)$ as follows:

$$z_0(i) = \max(\mu_z(i), z_0^{\rm floor}(i)) \quad (44)$$

where

$$z_0^{\rm floor}(i) = \frac{1}{20}\mathrm{SNR}^{\rm floor} - \frac{1}{2}\log\frac{\sigma^2_z(i)}{\sigma^2_n(i)} + n_0(i) \quad (45)$$

with $\mathrm{SNR}^{\rm floor}$ being a predefined threshold for $\mathrm{SNR}(i)$.

For high-order VTS approximation, no closed-form solution of $z_0(i)$ like (44) exists for a given $\mathrm{SNR}^{\rm floor}$. Based on (41) and (42), $\mathrm{SNR}(i)$ is related to $\sigma^2_z(i)$, $\sigma^2_y(i)$, and $\sigma^2_{zy}(i)$, which can be calculated as described in Section III. Consequently, we can identify a nonlinear and monotonic function $\mathrm{SNR}(i) = S(z_0(i))$. The problem becomes that given $\mathrm{SNR}^{\rm floor}$, how to find the corresponding constrained $z_0(i)$ by $S(\cdot)$. A numerical solution can be obtained by an iterative process as follows:

```
Step 1: Set $z_0(i) = \mu_z(i)$. Calculate $\mathrm{SNR}(i) = S(z_0(i))$.
Step 2: If $\mathrm{SNR}(i) \geq \mathrm{SNR}^{\rm floor}$, exit; otherwise, go to Step 3.
Step 3: Calculate $\Delta_z = |(1/20)\mathrm{SNR}^{\rm floor} - (1/2)\log(\sigma^2_z(i)/\sigma^2_n(i))|$ (inspired by (45)).
        Set $z_0(i) = \mu_z(i) + \Delta_z$.
        • While $(S(z_0(i)) < \mathrm{SNR}^{\rm floor})$ do $\{\Delta_z = 2\Delta_z; z_0(i) = \mu_z(i) + \Delta_z\}$.
Step 4: Set $z_0^{\rm left}(i) = \mu_z(i) + \Delta_z/2$, $z_0^{\rm right}(i) = \mu_z(i) + \Delta_z$. Obviously $S(z_0^{\rm left}(i)) < \mathrm{SNR}^{\rm floor}$
        and $S(z_0^{\rm right}(i)) \geq \mathrm{SNR}^{\rm floor}$. Given $z_0^{\rm left}(i)$ and $z_0^{\rm right}(i)$, we can then use bisection method to update $z_0(i)$
        iteratively until a maximum number of iterations (10 in our experiments) is reached or the following criterion is satisfied:
        $$\left|\frac{\mathrm{SNR}(i) - \mathrm{SNR}^{\rm floor}}{\mathrm{SNR}^{\rm floor}}\right| < 0.001.$$
```

The modified approach to MMSE-based clean speech estimation with the above safeguard measure will be referred to as "MMSE-SAFE" hereinafter.

As a special case, when $\boldsymbol{\Sigma}^{\rm c}_{\mathbf{zy},m}(\boldsymbol{\Sigma}^{\rm c}_{\mathbf{y},m})^{-1}$ is set directly to 1, the MMSE-based clean speech estimation using (20) and (21) becomes the same as what was described in [16], [17], which will be referred to as "MMSE-VTS0" approach hereinafter. Actually, this case corresponds to the zero-order VTS approximation described in [16] and [17].

V. Experiments and Results

The traditional first-order VTS-based feature compensation approach has been treated as an established technique for noisy speech recognition. In [16], VTS-based approach has been verified to outperform several noise reduction techniques, e.g., RATZ (Multivariate-Gaussian-Based Cepstral Normalization) and CDCN (Codeword Dependent Cepstrum Normalization). A comparison of the computational complexity of the first-order VTS approach with other techniques is also given there. Therefore, in this study, experiments are designed to compare the performance of our proposed HOVTS approaches with the first-order VTS approach only.

A. Experimental Setup

As a proof-of-concept study, Aurora2 [10] and Aurora4 [11], [18] databases are used to verify the effectiveness of the proposed approach for the small-vocabulary task of recognition of connected digit strings and the large-vocabulary continuous speech recognition (LVCSR) task, respectively. Both Aurora2 and Aurora4 databases contain speech data in the presence of additive noises and linear convolutional distortions, which were introduced synthetically to "clean" speech derived from TIDigits [15] and WSJ [19] databases, respectively. The modeling assumptions made in our adopted distortion model can be treated as correct in both cases. In order to verify the effectiveness of the proposed approach on real-world ASR, Aurora3 database [2]–[5] was used, which contains utterances of digit strings recorded in real automobile environments for Danish, Finnish, German and Spanish, respectively. A full description of the above databases and the corresponding test frameworks are given in [2]–[5], [10], [11], [18].

TABLE I
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF THE BASELINE SYSTEM AND VTS(N) SYSTEMS USING VTS-BASED FEATURE COMPENSATION, AVERAGED OVER SNRS BETWEEN 0 dB AND 20 dB ACROSS ALL NOISE CONDITIONS ON THREE DIFFERENT TEST SETS OF AURORA2 DATABASE

Methods		Set A	Set B	Set C
Baseline		66.36	71.43	67.20
VTS(N)	1st-order	86.21	85.24	82.65
	2nd-order	87.18	86.61	84.90
	3rd-order	**87.65**	**87.01**	**85.39**

In our ASR systems, the feature vector we used consists of 13 MFCCs (including C_0) plus their first- and second-order derivatives. The number of Mel-frequency filter banks is 23. MFCCs are computed based on power spectra. The mixture number of clean-speech GMM for feature compensation is 256. For Aurora2 and Aurora3 tasks, each digit is modeled by a whole-word left-to-right CDHMM, which consists of 16 emitting states, each having three Gaussian mixture components. For Aurora4 task, triphones are used as basic speech units. Each triphone is modeled by a CDHMM with three emitting states, each having eight Gaussian mixture components. There are in total 2800 tied states based on decision trees. A bigram language model (LM) for a 5 k-word vocabulary is used in recognition.

For experiments on Aurora2 and Aurora4 databases, "clean-training" is used, where "8 kHz data" is used for Aurora4. For Aurora3 experiments, we focus on high-mismatch (HM) "training-testing" condition, where training data includes utterances recorded by close-talking (CT) microphone, which can be considered as "clean," while testing data is recorded by hands-free (HF) microphone. For re-estimation of distortion model parameters, the control parameter N_{VTS} is set as 4. Our baseline ASR systems used CMN for feature compensation. In all the experiments, tools in HTK [22] are used for training and testing. In the following subsections, we report the experimental results.

Proof-of-Concept Study on Aurora2 and Aurora4 Tasks

In the first set of experiments, we study the effectiveness of one of our HOVTS-based feature compensation methods on Aurora2 database, where only additive noise is considered in our distortion model (i.e., the method in [8] and referred to as "VTS(N)" method hereinafter). Table I summarizes a performance (word accuracy in %) comparison of the baseline system and VTS(N) systems. The performance is averaged over SNRs between 0 dB and 20 dB on test Set A, Set B and Set C, respectively. Several observations can be made. First, all VTS(N) systems outperform the "Baseline" system. Higher the order in VTS approximation, better the performance. Second, although "3rd-order VTS(N)" achieves the best performance in all cases, the gap between "2nd-order VTS(N)" and "3rd-order VTS(N)" is small.

In the second set of experiments, we study the effectiveness of another HOVTS-based feature compensation method on Aurora2 database, where both additive noise and convolutional distortion are considered in our distortion model as described in Section II. This method is referred to as "VTS(N,H)" method hereinafter. Table II compares the performance of VTS(N) and VTS(N,H) systems on Set C, where both additive noise and channel mismatch exist. As expected, VTS(N,H) performs consistently better than VTS(N), because the channel mismatch was compensated for as well in VTS(N,H) method. Again, it is observed that higher the order in VTS approximation, better the performance. Considering the tradeoff between recognition performance and computational complexity, we will not use an order higher than 2 in the following experiments.

TABLE II
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF VTS(N) AND VTS(N,H) SYSTEMS USING VTS-BASED FEATURE COMPENSATION, AVERAGED OVER SNRS BETWEEN 0 dB AND 20 dB ACROSS ALL NOISE CONDITIONS ON TEST SET C OF AURORA2 DATABASE

	1st-order	2nd-order	3rd-order
VTS(N)	82.65	84.90	85.39
VTS(N,H)	**84.17**	**85.14**	**85.62**

TABLE III
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF THE BASELINE SYSTEM AND SEVERAL ROBUST ASR SYSTEMS USING VTS-BASED FEATURE COMPENSATION FOR THE SECOND MICROPHONE UNDER SEVERAL NOISY ENVIRONMENTS OF AURORA4 DATABASE

Methods		Babble	Restaurant	Street
Baseline		44.22	40.48	36.07
VTS(N)	1st-order	51.37	45.66	46.24
	2nd-order	55.61	48.51	51.02
VTS(N,H)	1st-order	56.59	47.45	51.28
	2nd-order	**57.58**	**50.85**	**52.05**

The above experiments are repeated on Aurora4 task. Table III summarizes a performance (word accuracy in %) comparison of the baseline system and several robust ASR systems using VTS-based feature compensation for the second microphone under several noisy environments of Aurora4 database. Due to the different microphones used in training (Sennheiser microphone) and testing (the second microphone), these results are used for demonstrating the effects of both additive noises and channel mismatch. The same observations can be made as on Aurora2 task.

From the above experimental results, it is clear that the proposed HOVTS-based feature compensation methods can improve the recognition accuracy under different conditions compared with its first-order VTS counterpart when the modeling assumption of our distortion model reflects the truth in unknown utterances to be recognized.

B. Effects on a Real-World ASR Task

In order to verify the effectiveness of the proposed approach on real-world ASR, a set of comparative experiments are conducted on Aurora3 database. Table IV summarizes a performance (word accuracy in %) comparison of the baseline system and the following robust ASR systems using VTS-based feature compensation in the high-mismatch (HM) condition on Aurora3 database:

- **VTS(N,H)(Standard)**: the approach described in Section II;
- **CMN+VTS(N,H)(Standard)**: the approach described in Section IV-A;
- **CMN+VTS(N,H)(MMSE-VTS0)**: the combined CMN and VTS-based feature compensation approach, where the MMSE-VTS0 approach as described in Section IV-B is used for clean speech estimation;

TABLE IV
PERFORMANCE (WORD ACCURACY IN %) COMPARISON OF THE BASELINE SYSTEM AND SEVERAL ROBUST ASR SYSTEMS USING VTS-BASED FEATURE COMPENSATION IN THE HIGH-MISMATCH (HM) CONDITION ON AURORA3 DATABASE

Methods		German	Danish	Finnish	Spanish
Baseline		83.77	54.78	77.07	80.96
VTS(N,H) (Standard)	1st-order	89.87	56.86	83.50	*77.83*
	2nd-order	90.01	63.40	84.81	*79.46*
CMN+VTS(N,H) (Standard)	1st-order	89.59	74.18	84.81	84.39
	2nd-order	89.96	72.14	85.16	84.27
CMN+VTS(N,H) (MMSE-VTS0)	1st-order	91.03	77.00	86.93	86.14
	2nd-order	90.98	76.92	87.53	87.13
CMN+VTS(N,H) (MMSE-SAFE)	1st-order	91.03	79.43	87.67	87.67
	2nd-order	**91.21**	**79.58**	**88.16**	**88.39**

- **CMN+VTS(N,H)(MMSE-SAFE)**: the combined CMN and VTS-based feature compensation approach, where the MMSE-SAFE approach as described in Section IV-B is used for clean speech estimation. The corresponding SNR thresholds for German, Danish, Finnish, and Spanish, which are determined on a development set, are set as 20, 7.5, 20, and 12.5 dB, respectively. The performance of MMSE-SAFE-based approach is not sensitive to these SNR thresholds beyond certain levels

From the above experimental results, we made the following observations.

- Although “VTS(N,H)(Standard)” approach performs better than the baseline system in most of cases, it indeed performs worse than the baseline system on Spanish task, especially for the first-order VTS case.
- All the combined CMN and VTS-based feature compensation approaches achieve significant performance improvement against the Baseline system for all the tasks. They also perform better than the “VTS(N,H)(Standard)” approach in most of cases.
- Both modified MMSE-based approaches for clean speech estimation help. MMSE-SAFE based approach is more complicated but achieves better performance than the MMSE-VTS0 approach.
- Overall, the “CMN+VTS(N,H)(MMSE-SAFE)” approach achieves the best performance. In this case, the second-order VTS approximation performs better than the first-order case.

C. Computational Complexity

Another concern of our proposed feature compensation solution is its computational complexity during recognition stage. The main overhead comes from the noise/channel estimation and the clean speech estimation, which are affected directly by the number (i.e., M) of components in clean-speech GMM and the number of EM iterations (i.e., N_{VTS}) in estimating distortion model parameters. Table V gives readers an idea of how the User CPU Time (in second) looks like for the above-mentioned two modules of the **CMN+VTS(N,H)(MMSE-SAFE)** approach. The timing experiment is conducted on a Pentium IV PC with a clock rate of 2.66 GHz by using a randomly selected testing sentence with a length of 1.54 s from Aurora3 database. The relevant control parameters are set as $M = 256$, $N_{VTS} = 4$. In practice, it is observed that a large portion of recognition accuracy improvement is still kept by decreasing the mixture number from 256 to 8 and the number of EM iterations from 4 to 2. The efficiency of our proposed method is not a big issue under this setting.

TABLE V
SUMMARY OF THE USER CPU TIME (IN SECONDS) FOR THE NOISE/CHANNEL ESTIMATION COMPONENT AND THE CLEAN SPEECH ESTIMATION COMPONENT OF THE **CMN+VTS(N,H)(MMSE-SAFE)** APPROACH BY USING A RANDOMLY SELECTED TESTING SENTENCE WITH A LENGTH OF 1.54 s FROM AURORA3 DATABASE ($M = 256, N_{VTS} = 4$)

Configuration	1st-order	2nd-order
noise/channel estimation	1.96	5.82
clean speech estimation	0.65	2.25

VI. CONCLUSION AND DISCUSSIONS

In this paper, we have presented a new feature compensation approach using high-order vector Taylor series (HOVTS) approximation of an explicit distortion model. Experimental results on Aurora2 and Aurora4 benchmark databases, where the modeling assumption of the distortion model is more accurate, demonstrate that the standard HOVTS-based feature compensation approaches achieve consistently significant improvement in recognition accuracy compared to traditional standard first-order VTS based approach. For a real-world in-vehicle connected digits recognition task on Aurora3 benchmark database where the modeling assumption of the distortion model is less accurate, modifications are necessary to make VTS-based feature compensation approaches work. In this case, the second-order VTS-based approach performs only slightly better than the first-order VTS based approach. By further considering computational complexity of different approaches, the “CMN+VTS(N,H)(MMSE-VTS0)” approach using the first-order VTS approximation in distortion model parameter estimation offers a very attractive practical solution. We therefore recommend our readers to try out this approach in their applications.

APPENDIX
DERIVATION OF ML TRAINING OF DISTORTION MODEL PARAMETERS

In this appendix, we summarize how to derive, by extending the formulations in, e.g., [14] and [20], a procedure for the estimation of the parameters of explicit distortion model by maximizing the likelihood function defined on a given set of noisy observations in cepstral domain.

First we make a general assumption that both $\mathbf{z}$ and $\mathbf{n}$ are modeled by GMMs, although only a single Gaussian model is used for additive noise vector $\mathbf{n}$ in this paper. The likelihood function is defined as

$$\begin{aligned}\mathcal{L}(\mathbf{Y}|\mathbf{\Lambda}) &= p(\mathbf{Y}|\mathbf{\Lambda}) = p(\mathbf{Y}|\mathbf{\Lambda_z}, \mathbf{\Lambda_n}) \\ &= \sum_{\mathbf{M_z}} \sum_{\mathbf{M_n}} p(\mathbf{Y}, \mathbf{M_z}, \mathbf{M_n}|\mathbf{\Lambda_z}, \mathbf{\Lambda_n}) \qquad (46)\end{aligned}$$

where $\mathbf{\Lambda}_z$ and $\mathbf{\Lambda}_n$ are model parameter sets for $\mathbf{z}$ and $\mathbf{n}$, respectively. $\mathbf{Y}$ is the sequence of the noisy observation vectors in the current utterance. $\mathbf{M_z}$ and $\mathbf{M_n}$ are the sequences of Gaussian component indices for $\mathbf{z}$ and $\mathbf{n}$, respectively. $p(\mathbf{Y},\mathbf{M_z},\mathbf{M_n}|\mathbf{\Lambda_z},\mathbf{\Lambda_n})$ can be expressed as

$$\begin{aligned}&p(\mathbf{Y},\mathbf{M_z},\mathbf{M_n}|\mathbf{\Lambda_z},\mathbf{\Lambda_n})\\&=\iint_C \prod_{t=1}^{T}\omega_{\mathbf{z}}(m_{\mathbf{z}}^t)\omega_{\mathbf{n}}(m_{\mathbf{n}}^t)p(\mathbf{z}^t|m_{\mathbf{z}}^t)p(\mathbf{n}^t|m_{\mathbf{n}}^t)d\mathbf{Z}d\mathbf{N}\end{aligned} \tag{47}$$

where $m_{\mathbf{z}}^t$ and $m_{\mathbf{n}}^t$ are the hidden Gaussian component indices at the tth frame for $\mathbf{z}$ and $\mathbf{n}$, respectively; $\omega_{\mathbf{z}}(m_{\mathbf{z}}^t)$ and $\omega_{\mathbf{n}}(m_{\mathbf{n}}^t)$ denote the weights of the corresponding Gaussian components for $\mathbf{z}^t$ and $\mathbf{n}^t$, respectively; $p(\mathbf{z}^t|m_{\mathbf{z}}^t)$ and $p(\mathbf{n}^t|m_{\mathbf{n}}^t)$ are pdfs of Gaussian components for $\mathbf{z}$ and $\mathbf{n}$, respectively; and the notation $\iint_C$ represents the T-fold iterated integral, each component of which is along the contour C_t defined by the explicit model $f(\mathbf{z}^t,\mathbf{n}^t)=\mathbf{y}^t$. It is important to note that one can define a particular model for the corruption of the clean speech by noise simply by defining particular contours of integration C_t.

It is impossible to obtain the closed-form ML estimation directly by maximizing the likelihood function in (46). Here we adopt an iterative EM algorithm to solve the problem. The M-Step of the EM algorithm is to maximize the following auxiliary function:

$$\begin{aligned}\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})&=E[\log p(\mathbf{Z},\mathbf{N},\mathbf{M_z},\mathbf{M_n}|\bar{\mathbf{\Lambda}})|\mathbf{Z},\mathbf{N},\mathbf{\Lambda}]\\&=\sum_{\mathbf{M_z}}\sum_{\mathbf{M_n}}\iint_C p(\mathbf{Z},\mathbf{N},\mathbf{M_z},\mathbf{M_n}|\mathbf{\Lambda})\\&\quad\log p(\mathbf{Z},\mathbf{N},\mathbf{M_z},\mathbf{M_n}|\bar{\mathbf{\Lambda}})d\mathbf{Z}d\mathbf{N}\end{aligned} \tag{48}$$

where

$$p(\mathbf{Z},\mathbf{N},\mathbf{M_z},\mathbf{M_n}|\mathbf{\Lambda})=\prod_{t=1}^{T}\omega_{\mathbf{z}}(m_{\mathbf{z}}^t)\omega_{\mathbf{n}}(m_{\mathbf{n}}^t)p(\mathbf{z}^t|m_{\mathbf{z}}^t)p(\mathbf{n}^t|m_{\mathbf{n}}^t) \tag{49}$$

and $\mathbf{\Lambda}$ and $\bar{\mathbf{\Lambda}}$ are the sets of old and new model parameters, respectively. If we assume that the observations are independent in time and random processes representing $\mathbf{Z},\mathbf{N},\mathbf{M_z},\mathbf{M_n}$, are independent, $\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})$ can be reduced to

$$\begin{aligned}\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})&=\sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\iint_C \gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})\\&\quad\log\left[\bar{\omega}_{\mathbf{z}}(k_{\mathbf{z}})\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})\bar{p}(\mathbf{z}^t|k_{\mathbf{z}})\bar{p}(\mathbf{n}^t|k_{\mathbf{n}})\right]d\mathbf{Z}d\mathbf{N}\end{aligned} \tag{50}$$

where

$$\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})=\sum_{\mathbf{M_z}}\sum_{\mathbf{M_n}}\delta(k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{M_z},\mathbf{M_n})p(\mathbf{Z},\mathbf{N},\mathbf{M_z},\mathbf{M_n}|\mathbf{\Lambda}); \tag{51}$$

$k_{\mathbf{z}}$ and $k_{\mathbf{n}}$ are the Gaussian component indices for $\mathbf{z}$ and $\mathbf{n}$, respectively; and $\delta(k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{M_z},\mathbf{M_n})$ is an indicator function defined as follows:

$$\delta(k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{M_z},\mathbf{M_n})=\begin{cases}1, & \text{if } m_{\mathbf{z}}^t=k_{\mathbf{z}}, m_{\mathbf{n}}^t=k_{\mathbf{n}}\\0, & \text{otherwise.}\end{cases} \tag{52}$$

Individually maximizing $\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})$ in (50) with respect to each of the model parameters in $\bar{\mathbf{\Lambda}}_{\mathbf{n}}$ is straightforward. Maximizing (50) with respect to $\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})$ under the constraint $\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})=1$ gives

$$\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})=\frac{\sum\limits_{t=1}^{T}\sum\limits_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\iint_C\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})d\mathbf{Z}d\mathbf{N}}{\sum\limits_{t=1}^{T}\sum\limits_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\sum\limits_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\iint_C\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})d\mathbf{Z}d\mathbf{N}}. \tag{53}$$

Meanwhile, it is easy to prove that

$$\iint_C\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})d\mathbf{Z}d\mathbf{N}=p(\mathbf{Y}|\mathbf{\Lambda})p(k_{\mathbf{z}},k_{\mathbf{n}}|\mathbf{y}^t,\mathbf{\Lambda}) \tag{54}$$

where

$$p(k_{\mathbf{z}},k_{\mathbf{n}}|\mathbf{y}^t,\mathbf{\Lambda})=\frac{\omega_{\mathbf{z}}(k_{\mathbf{z}})\omega_{\mathbf{n}}(k_{\mathbf{n}})p(\mathbf{y}^t|k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{\Lambda})}{\sum\limits_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\sum\limits_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\omega_{\mathbf{z}}(k_{\mathbf{z}})\omega_{\mathbf{n}}(k_{\mathbf{n}})p(\mathbf{y}^t|k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{\Lambda})} \tag{55}$$

is the posterior probability of two hidden random variables $k_{\mathbf{z}}$ and $k_{\mathbf{n}}$. Substituting (54) into (53), the final updating formula for $\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})$ is as follows:

$$\bar{\omega}_{\mathbf{n}}(k_{\mathbf{n}})=\frac{1}{T}\sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}p(k_{\mathbf{z}},k_{\mathbf{n}}|\mathbf{y}^t,\mathbf{\Lambda}). \tag{56}$$

The mean and covariance matrix of Gaussian components can be estimated similarly. By setting the following partial derivative

$$\begin{aligned}&\frac{\partial\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})}{\partial\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})}\\&=\sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}}\iint_C\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})\frac{\partial\log\bar{p}(\mathbf{n}^t|k_{\mathbf{n}})}{\partial\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})}d\mathbf{Z}d\mathbf{N}\end{aligned} \tag{57}$$

equal to zero with the Gaussian pdf $\bar{p}(\mathbf{n}^t|k_{\mathbf{n}})$, we have

$$\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})=\frac{\sum\limits_{t=1}^{T}\sum\limits_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\iint_C\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})\mathbf{n}^t d\mathbf{Z}d\mathbf{N}}{\sum\limits_{t=1}^{T}\sum\limits_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\iint_C\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})d\mathbf{Z}d\mathbf{N}}. \tag{58}$$

The integral in the numerator of (58) can be reduced to

$$\begin{aligned}&\iint_C\gamma_t(k_{\mathbf{z}},k_{\mathbf{n}})\mathbf{n}^t d\mathbf{Z}d\mathbf{N}\\&=p(\mathbf{Y}|\mathbf{\Lambda})p(k_{\mathbf{z}},k_{\mathbf{n}}|\mathbf{y}^t,\mathbf{\Lambda})E_{\mathbf{n}}[\mathbf{n}^t|\mathbf{y}^t,k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{\Lambda}]\end{aligned} \tag{59}$$

where $E_{\mathbf{n}}[\mathbf{n}^t|\mathbf{y}^t,k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{\Lambda}]$ is the conditional expectation of $\mathbf{n}^t$ given $\mathbf{y}^t$ for components $k_{\mathbf{z}}$ and $k_{\mathbf{n}}$. By substituting (54) and (59) into (58), the updating formula for $\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})$ can be obtained as

$$\bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})=\frac{\sum\limits_{t=1}^{T}\sum\limits_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}p(k_{\mathbf{z}},k_{\mathbf{n}}|\mathbf{y}^t,\mathbf{\Lambda})E_{\mathbf{n}}[\mathbf{n}^t|\mathbf{y}^t,k_{\mathbf{z}},k_{\mathbf{n}},\mathbf{\Lambda}]}{\sum\limits_{t=1}^{T}\sum\limits_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}p(k_{\mathbf{z}},k_{\mathbf{n}}|\mathbf{y}^t,\mathbf{\Lambda})}. \tag{60}$$

By a similar procedure, the updating formula for the covariance matrix $\bar{\mathbf{\Sigma}}_{\mathbf{n}}(k_{\mathbf{n}})$ can also be obtained as follows:

$$\bar{\mathbf{\Sigma}}_{\mathbf{n}}(k_{\mathbf{n}}) = \frac{\sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}} p(k_{\mathbf{z}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda}) E_{\mathbf{n}}[\mathbf{n}^t(\mathbf{n}^t)^\top|\mathbf{y}^t, k_{\mathbf{z}}, k_{\mathbf{n}}, \mathbf{\Lambda}]}{\sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}} p(k_{\mathbf{z}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})} - \bar{\boldsymbol{\mu}}_{\mathbf{n}}(k_{\mathbf{n}})\bar{\boldsymbol{\mu}}_{\mathbf{n}}^\top(k_{\mathbf{n}}) \quad (61)$$

where $E_{\mathbf{n}}[\mathbf{n}^t(\mathbf{n}^t)^\top|\mathbf{y}^t, k_{\mathbf{z}}, k_{\mathbf{n}}, \mathbf{\Lambda}]$ is the conditional expectation of $\mathbf{n}^t(\mathbf{n}^t)^\top$ given $\mathbf{y}^t$ for components $k_{\mathbf{z}}$ and $k_{\mathbf{n}}$.

Now let us consider how to derive the updating formula for $\mathbf{h}_{\text{const}}$. Given that $\mathbf{z} = \mathbf{x} + \mathbf{h}$ and $p(\mathbf{h}) = \delta(\mathbf{h} - \mathbf{h}_{\text{const}})$, we have

$$\bar{p}(\mathbf{z}^t|k_{\mathbf{z}}) = \mathcal{N}(\mathbf{z}^t; \boldsymbol{\mu}_{\mathbf{x}}(k_{\mathbf{z}}) + \bar{\mathbf{h}}_{\text{const}}, \mathbf{\Sigma}_{\mathbf{x}}(k_{\mathbf{z}}))\ . \quad (62)$$

The partial derivative of $\mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})$ in (50) with respect to $\bar{\mathbf{h}}_{\text{const}}$ can be written as

$$\frac{\partial \mathcal{Q}(\bar{\mathbf{\Lambda}}|\mathbf{\Lambda})}{\partial \bar{\mathbf{h}}_{\text{const}}} = \sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}}\sum_{k_{\mathbf{n}}=1}^{K_{\mathbf{n}}} \int\!\!\int_C \gamma_t(k_{\mathbf{z}}, k_{\mathbf{n}}) \frac{\partial \log \bar{p}(\mathbf{z}^t|k_{\mathbf{z}})}{\partial \bar{\mathbf{h}}_{\text{const}}} d\mathbf{Z}d\mathbf{N}. \quad (63)$$

Using (62) and setting the above expression equal to zero, the updating formula for $\bar{\mathbf{h}}_{\text{const}}$ can be derived as

$$\bar{\mathbf{h}}_{\text{const}} = \left[\sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}} p(k_{\mathbf{z}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})\mathbf{\Sigma}_{\mathbf{x}}^{-1}(k_{\mathbf{z}})\right]^{-1} \sum_{t=1}^{T}\sum_{k_{\mathbf{z}}=1}^{K_{\mathbf{z}}} \Big[p(k_{\mathbf{z}}, k_{\mathbf{n}}|\mathbf{y}^t, \mathbf{\Lambda})\mathbf{\Sigma}_{\mathbf{x}}^{-1}(k_{\mathbf{z}}) \big(E_{\mathbf{z}}[\mathbf{z}^t|\mathbf{y}^t, k_{\mathbf{z}}, k_{\mathbf{n}}, \mathbf{\Lambda}] - \boldsymbol{\mu}_{\mathbf{x}}(k_{\mathbf{z}})\big)\Big] \quad (64)$$

where $E_{\mathbf{z}}[\mathbf{z}^t|\mathbf{y}^t, k_{\mathbf{z}}, k_{\mathbf{n}}, \mathbf{\Lambda}]$ is the conditional expectation of $\mathbf{z}^t$ given $\mathbf{y}^t$ for components $k_{\mathbf{z}}$ and $k_{\mathbf{n}}$.

REFERENCES

[1] A. Acero, *Acoustic and Environment Robustness in Automatic Speech Recognition*. Norwell, MA: Kluwer, 1993.

[2] "Availability of Finnish SpeechDat-Car Database for ETSI STQ WI008 Front-End Standardization," Nokia, 1999, Aurora document AU/217/99.

[3] "Spanish SDC-Aurora Database for ETSI STQ Aurora WI008 Advanced DSR Front-End Evaluation: Description and Baseline Results," UPC, 2000, Aurora document AU/271/00.

[4] "Description and Baseline Results for the Subset of the SpeechDat-Car German Database Used for ETSI STQ Aurora WI008 Advanced DSR Front-End Evaluation," Texas Instruments, 2001, Aurora doc. AU/273/00.

[5] "Danish SpeechDat-Car Digits Database for ETSI STQ-Aurora Advanced DSR," Aalborg Univ., 2001, Aurora document AU/378/01.

[6] G.-H. Ding and B. Xu, "Exploring high-performance speech recognition in noisy environments using high-order Taylor series expansion," in *Proc. ICSLP*, 2004, pp. 149–152.

[7] J. Du and Q. Huo, "Feature Compensation Using High-Order Vector Taylor Series for Noisy Speech Recognition," Tech. Memo MSRA, 2008.

[8] 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, pp. 1257–1260.

[9] J. Du, Q. Huo, and Y. Hu, "Evaluation of a feature compensation approach using high-order vector Taylor series approximation of an explicit distortion model on Aurora2, Aurora3, and Aurora4 tasks," in *Proc. ISCSLP*, 2008, pp. 81–84.

[10] 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, pp. 181–188.

[11] H. G. Hirsch, "Experimental Framework for the Performance Evaluation of Speech Recognition Front-Ends on a Large Vocabulary Task, Version 2.0," ETSI STQ-Aurora DSR Working Group, 2002.

[12] Y. Hu and Q. Huo, "Irrelevant variability normalization based HMM training using VTS approximation of an explicit model of environmental distortions," in *Proc. Interspeech*, 2007, pp. 1042–1045.

[13] N. S. Kim, "Statistical linear approximation for environment compensation," *IEEE Signal Process. Lett.*, vol. 5, no. 1, pp. 8–10, 1998.

[14] D.-Y. Kim, C.-K. Un, and N.-S. Kim, "Speech recognition in noisy environments using first-order vector Taylor series," *Speech Commun.*, vol. 24, pp. 39–49, 1998.

[15] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proc. ICASSP*, 1984, pp. 42.11.1–42.11.4.

[16] P. J. Moreno, "Speech recognition in noisy environments," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, 1996.

[17] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. ICASSP*, 1996, pp. 733–736.

[18] N. Parihar and J. Picone, "DSR Front End LVCSR Evaluation," Aurora Working Group, ETSI, 2002, AU/384/02.

[19] D. Paul and J. Baker, "The design of Wall Street Journal-based CSR corpus," in *Proc. ICSLP*, 1992, pp. 899–902.

[20] R. C. Rose, E. M. Hofstetter, and D. A. Reynolds, "Integrated models of signal and background with application to speaker identification in noise," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 2, pp. 245–257, Apr. 1994.

[21] V. Stouten, "Robust automatic speech recognition in time-varying environemnts," Ph.D. dissertation, Katholieke Univ. Leuven, Leuven, Belgium, 2006.

[22] S. Young *et al., The HTK Book (for HTK v3.4)*. Cambridge, U.K.: Cambridge Univ. Eng. Dept., 2006.

Jun Du received the B.Eng. and Ph.D. degrees from the Department of Electronic Engineering and Information Science, University of Science and Technology of China (USTC), Hefei, in 2004 and 2009, respectively.

From 2004 to 2009, he was with the iFlytek Speech Lab, USTC, where he conducted research on speech recognition. During the above period, he worked as an Intern twice for 9 months at Microsoft Research Asia (MSRA), Beijing, China, doing research on discriminative training and noise-robust front-end for speech recognition, and speech enhancement. In 2007, he also worked as a Research Assistant for 6 months at the Department of Computer Science, The University of Hong Kong, doing research on robust speech recognition. From July 2009 to June 2010, he worked at iFlytek Research on speech recognition. In July 2010, he joined the Visual Computing Group, MSRA, as an Associate Researcher.

Qiang Huo (M'95) received the B.Eng. degree from the University of Science and Technology of China (USTC), Hefei, in 1987, the M.Eng. degree from Zhejiang University, Hangzhou, China, in 1989, and the Ph.D. degree from the USTC in 1994, all in electrical engineering.

He is a Lead Researcher and Research Manager with Microsoft Research Asia (MSRA), Beijing, China. Prior to joining MSRA on August 1, 2007, he had been a faculty member at the Department of Computer Science, The University of Hong Kong since 1998, where he also did his Ph.D. research on speech recognition during 1991 to 1994. From 1995 to 1997, he worked at the ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan. In the past 25 years, he has been doing research and making contributions in the areas of speech recognition, handwriting recognition, OCR, gesture recognition, biometric-based user authentication, hardware design for speech, and image processing.