Author(s): Kalle J. Palomaki, Heikki Kallasjoki

Title: Reverberation robust speech recognition by matching distributions of spectrally and temporally decorrelated features

Year: 2014

Version: Final published version

**Please cite the original version:**

Kalle J. Palomaki, Heikki Kallasjoki. Reverberation robust speech recognition by matching distributions of spectrally and temporally decorrelated features. In Proceedings of the REVERB Workshop, Florence, Italy, May 2014

# REVERBERATION ROBUST SPEECH RECOGNITION BY MATCHING DISTRIBUTIONS OF SPECTRALLY AND TEMPORALLY DECORRELATED FEATURES

*Kalle J. Palomäki and Heikki Kallasjoki*

Department of Signal Processing and Acoustics, Aalto University, Finland;
e-mail: {kalle.palomaki,heikki.kallasjoki}@aalto.fi.

## ABSTRACT

This paper addresses dereverberation of speech using an unsupervised approach utilizing speech prior and taking only weak assumptions on reverberation. Our approach uses a long time context representation of reverberated speech in spectral-temporal supervectors which are decorrelated by the PCA. In the decorrelated domain supervectors are mapped from reverberant speech distribution to clean speech distribution and then to mel-spectral vectors. Mel-domain Wiener filter is applied as post processing. Our results demonstrate performance gains over the provided baseline recognizer, and show that the method can be coupled to CMLLR adaptation with cumulative benefits for clean trained models. Furthermore, we show that using dimensionality reduction coupled with the Wiener filter is better than using full dimensional PCA in representing small variance components in speech.

***Index Terms***— dereverberation, speech recognition, supervector, decorrelation, unsupervised

## 1. INTRODUCTION

Considering automatic speech recognizers (ASR) used in practical applications we cannot often control recording conditions but are reliant on hardware which end-users have. The quality of microphones, recording environments and distance between speaker and microphone can vary a great deal. The very same ASR system may need to cope with data from a single distant microphone, sophisticated arrays and close-talk microphones. This calls for methods that are unsupervised and not reliant on prior information about environments, arrays or specific microphones.

The conventional method to counteract reverberation or transmission line has been to produce robust features using cepstral mean normalization [1], modulation filtered spectrograms [2] or frequency domain linear prediction [3]. Their advantage is simplicity and wide applicability since they take only weak prior assumptions on the data, but used alone they yield only modest performance gains. Other approaches that can be used to improve reverberation robustness and that take only weak assumptions are missing feature methods with masks designed for reverberation [4, 5] or simply unsupervised adaptation [6, 7]. The advantage of Bayesian dereverberation approaches over the above mentioned, is that they can flexibly utilize either coarse or more precise source (speech) and filter (reverberation) models jointly [8, 9]. However, precise modeling of filter and source require computationally expensive methods such as Monte Carlo Markov Chain [9].

Almost regardless of the enhancement method, in practical ASR it is often common to utilize an adaptation methods as the last step to counteract variations from speakers or environments. Adaptation can be used based on the acoustic model distributions in ASR [6], and also based on data distributions using powerful non-linear distribution matching methods independently of acoustic models [7, 10]. The non-linear adaptation can also be combined with the acoustic model domain adaptation with cumulative benefits [7, 10].

The purpose of the present study is to develop a new distribution matching or adaptation method that is suitable to dereverberation. Due to long lasting effects of reverberation we utilize decorrelated spectral-temporal vectors that have some time context. Near relatives of the present study are non-linear adaptation [7] and feature space gaussianization [10]. The present study extends the previous work specifically addressing the problem of dereverberation, where as the previous work dealt primarily with speaker adaptation [7] or general purpose feature space gaussianization [10], the latter producing features that are easy to model with Gaussian mixtures with no specific intention to speech enhancement. Focusing on dereverberation leads us to use longer feature contexts. We also utilize post filtering methods that were not addressed in the above mentioned previous studies. Furthermore, we discuss the mathematical motivation why the proposed method is suitable for dereverberation.

## 2. DEREVERBERATION METHOD

Dereverberation can be considered as a Bayesian inverse problem in which an attempt is made to recover clean speech

TEKES FuNeSoMo (KJP), and by the Academy of Finland grants 136209 (KJP), and 251170 (KJP, HK, Finnish Centre of Excellence program 2012-2017) the Hecse graduate school (HK). The recognition experiments presented in this work were performed in part using computational resources within the Aalto University "Science-IT" project.

spectra $\boldsymbol{o_x}$ given noisy speech spectra $\boldsymbol{o_y}$. Posterior distribution for dereverberated speech $p(\boldsymbol{o_x}|\boldsymbol{o_y})$ is then

$$p(\boldsymbol{o_x}|\boldsymbol{o_y}) \approx p(\boldsymbol{o_x})p(\boldsymbol{o_y}|\boldsymbol{o_x}) \quad (1)$$

where $p(\boldsymbol{o_x})$ is the clean speech prior and $p(\boldsymbol{o_y}|\boldsymbol{o_x})$ represents the reverberant observation.

In signal processing terms reverberation can be considered as convolutive interference by reasonable accuracy. The convolution for the time domain speech signal $\mathsf{o}(t)$ and FIR filter h can be expressed as the matrix operation

$$\mathbf{b} = \mathbf{H}\mathbf{o} \quad (2)$$

where $\mathbf{H}$ is the Toeplitz matrix that represents the filter h.

Similarly we can express convolution in feature domains using linear transformations. For linear spectral features we can use matrix multiplication for expressing convolution. First, a supervector $\boldsymbol{s}(t) = [\boldsymbol{o}(t)^T \ \dots \ \boldsymbol{o}(t+T-1)^T]^T$ is formed from concatenation of $T$ consecutive frames of feature vectors, where $T$ is chosen large enough considering the length of the room impulse response. Here $\boldsymbol{o}$ without a subscript index such as $x$ or $y$ denotes arbitrary speech spectra, either clean or noisy. The dimensionality of the supervector is $N = TK$ where $K$ is the dimension of original features $\boldsymbol{o}$. From now on we drop the time index $t$ to keep notation simpler. The speech features $\boldsymbol{s_y}$ that are affected by convolution can be given as

$$\boldsymbol{s_y} = \boldsymbol{H_y}\boldsymbol{s_x} \quad (3)$$

where $\boldsymbol{H_y}$ represents the time invariant room impulse response and $\boldsymbol{s_x}$ represents clean speech.

Linear transformation $\boldsymbol{D}$ such as principal component analysis (PCA) can be applied to decorrelate rows of original supervectors $\boldsymbol{s}$

$$\boldsymbol{c} = \boldsymbol{D}\boldsymbol{H}\boldsymbol{s} \quad (4)$$

which allows treating elements of $\boldsymbol{c}$ one-by-one. Here $\boldsymbol{H}$ denotes an arbitrary impulse response. Now we denote $\boldsymbol{c}_i = \boldsymbol{D}\boldsymbol{H}_i\boldsymbol{s}$ and $\boldsymbol{c}_j = \boldsymbol{D}\boldsymbol{H}_j\boldsymbol{s}$ for the same supervector $\boldsymbol{s}$ but two different impulse responses $\boldsymbol{H}_i$ and $\boldsymbol{H}_j$. As the system is linear, we can transform $\boldsymbol{c}_i = \boldsymbol{A}\boldsymbol{c}_j$. Now we note that assuming $\boldsymbol{A}$ diagonal equals to assuming that matrix $\boldsymbol{D}$ decorrelates both $\boldsymbol{c}_i$ and $\boldsymbol{c}_j$. Then we can simplify the mapping of $\boldsymbol{c}_j(n)$, with its element indexed by $n = \{1, ..., N\}$, to multiplication by constant diagonal elements $a_{nn}$ of $\boldsymbol{A}$ as $c_i(n) \approx a_{nn}c_j(n)$.

For speech data, however, PCA is usually applied after logarithmic non-linearity

$$\boldsymbol{c'} = \boldsymbol{D'}\log\langle\boldsymbol{H}\boldsymbol{s}\rangle \quad (5)$$

where $\log\langle\ \rangle$ is computed element-wise. In this non-linear case we can transform $\boldsymbol{c'}_i = F(\boldsymbol{c'}_j)$, assuming that there is a bijective non-linear transformation $F$. By again assuming that matrix $\boldsymbol{D'}$ decorrelates both $\boldsymbol{c'}_i$ and $\boldsymbol{c'}_j$ we can use simpler element-wise mapping $F^{(n)}$ for $c'(n)$ so that $c'_i(n) \approx F^{(n)}(c'_j(n))$.

Given the decorrelated data, from now on we assume to work with component-wise data and drop corresponding indexes. For developing the mapping $F$ we apply a distribution matching method similar to [7, 10]. Empirical cumulative distribution $\Phi$ can be approximated by

$$\Phi(c') = \frac{1}{L}\sum_{k=1}^{L}\theta(c' - c'_k) \quad (6)$$

where $\theta$ is step function over $L$ samples of form $c'_k$ drawn from the distribution. It follows that simply sorting and scaling data gives and approximation of its inverse cumulative distribution function (ICDF). In our case we approximate reverberant speech posterior $p(c'_y|c'_x)$ and clean speech prior $p(c'_x)$ by samples of corresponding data, respectively. The samples are sorted component-wise , and we denote the resulting ICDF of clean speech sample by $\Phi_x^{-1}$ and reverberant speech by $\Phi_y^{-1}$. Finally we implement $F$ by constructing lookup table from $\Phi_y^{-1} \underset{F}{\rightarrow} \Phi_x^{-1}$ using Matlab `interp1` with piecewise cubic interpolation.

After the lookup table is defined, reverberant data can be transformed to dereverberated estimate of spectral supervector $\tilde{\boldsymbol{s}}$ in (log domain) by

$$\tilde{\boldsymbol{s}}' = \boldsymbol{D'}^{-1}F(\boldsymbol{c'}_y) \quad (7)$$

where mapping $F$ is applied element by element to reverberant decorrelated vectors $\boldsymbol{c'}_y$, and $\boldsymbol{D'}^{-1}$ inverts PCA to get back to log spectral-temporal domain. Then the log spectral supervector representation is dismantled to get an estimate of dereverberated speech log mel-spectrogram $\tilde{\boldsymbol{o_x}}'$. Each vector $\tilde{\boldsymbol{o_x}}'$ is obtained from supervector taking simply averages of parts of the adjacent supervector that correspond to it.

Then for $\tilde{\boldsymbol{o_x}}$ (expressed in linear domain) we apply a mel-spectral domain Wiener filter that is common in speech enhancement systems [11]. We define a mel-spectral domain Wiener filter $\boldsymbol{h_w}$ as

$$\boldsymbol{h_w} = \tilde{\boldsymbol{o_x}} \,./ \tilde{\boldsymbol{o_y}} \quad (8)$$

where $./$ denotes element-wise division and $\tilde{\boldsymbol{o_y}}$ represents reverberant data that has gone through the same PCA transformation $\boldsymbol{D'}$ and dimensionality reduction as dereverberated data (before lookup table mapping). The enhanced mel-spectral features are then obtained as

$$\hat{\boldsymbol{o_x}} = \boldsymbol{h_w} .* \boldsymbol{o_y}. \quad (9)$$

In the logarithmic domain ($\boldsymbol{o'} = \log(\boldsymbol{o})$) this can be written as

$$\hat{\boldsymbol{o_x}}' = \tilde{\boldsymbol{o_x}}' + \boldsymbol{o_y}' - \tilde{\boldsymbol{o_y}}' \quad (10)$$

where it can be seen more clearly that the filter sums back some of the variation in the reverberant signals through the

residual term $\boldsymbol{o_y}' - \tilde{\boldsymbol{o_y}}'$ that is absent in dereverberated $\tilde{\boldsymbol{o_x}}'$ smoothed by low order PCA. Now that we have the initial estimate of dereverberated speech $\tilde{\boldsymbol{o_x}}$ we apply the same process defined in equations (eq. 3) to (eq. 10) iteratively. In our case we use two iterations. Finally after two iterations the resulting $\hat{\boldsymbol{o_x}}$ are transformed to final features for the speech recognition.

## 3. EXPERIMENTS

This section describes the experimental evaluations of the system using data (Sect. 3.1) and baseline recognizers (Sect. 3.2) provided by the REVERB challenge. Parameters settings of the proposed approach (Sect. 3.2) and finally the results (Sect. 3.3) are also shown.

### 3.1. Data

The reverberant speech feature enhancement methods described in this work are evaluated on both artificially distorted clean speech ("SimData") and speech recorded in a noisy, reverberant room ("RealData"). Both data sets are provided by the REVERB challenge, and described in detail in [12]. Separate development and evaluation subsets are provided.

For SimData, clean speech utterances from the WSJCAM0 British English continuous speech recognition corpus [13] are first distorted using measured room impulse responses, and then mixed with measured room noise with a fixed signal-to-noise ratio (SNR) of 20 dB. Utterances in six simulated reverberant environments are provided: two speaker-to-microphone distances (near, far) in each of three rooms of varying size (small, medium, large). The near and far microphone distances are 0.5 m and 2.0 m, while $T_{60}$ reverberation times for the small, medium and large rooms were 0.25 s, 0.5 s and 0.7 s, respectively. The total number of utterances is 1484 and 2176 for the development and evaluation subsets, respectively.

The RealData set consists of real recordings of speakers in a reverberant meeting room. Contents of the utterances are based on the prompts of the WSJCAM0 corpus. The set contains two different test conditions, corresponding to near and far microphone distances of 1.0 m and 2.5 m, respectively. There are, respectively, 179 and 372 utterances in the development and evaluation subsets of RealData.

### 3.2. Speech Recognition System and Settings

The baseline recognizer provided by the REVERB challenge, based on the HTK toolkit [14], is used to evaluate the speech recognition performance of the proposed methods. The recognition system uses 13-dimensional Mel-frequency cepstral coefficients (MFCCs) augmented with first and second time derivatives. Hidden Markov models with 10-component Gaussian mixture emission distributions are used to model the acoustic features. The clean speech training set of the WSJCAM0 corpus [13] is used to train the acoustic models.

Unsupervised constrained MLLR (CMLLR) adaptation is optionally applied during recognition. For each test condition (room and recording distance), adaptation coefficients for 256 regression classes are calculated based on the entire test set. Unadapted recognition results are used to provide transcriptions for the adaptation.

The proposed distribution matching (DM) method uses spectral supervector representation (eq. 3) that is based on $T = 20$ or $T = 1$ frames of time context of $K = 23$ dimensional mel-spectrum computed in the process that is used generating final MFCC features in HTK. The mel spectral features are also treated with normalization method proposed in [4] to reduce effects spectral and gain alteration due to reverberation. PCA transformation (eq. 5) is estimated for clean speech data taken from the training part of corpus for over 1000 utterances. The primary approach using $T = 20$ time context uses $40$ principal components and version of approach using no time context ($T = 1$) uses 12 principal components. The full $460$ dimensional test version of the PCA uses the default context length $T = 20$. In the recognition phase, unless otherwise noted, we assume full batch processing and always collect reverberant posterior distributions (eq. 6) for the whole evaluation or development test condition, and for the corresponding speech prior we use equal length sample taken from the clean training set.

### 3.3. Results

Tables 1 and 2 show the results of the evaluation test on the clean and reverberant data, respectively. For the *reverberant* data (Table 2) ranking of the systems from poorest to the best based on averages is as follows: baseline without adaptation (Baseline), baseline with adaptation (Baseline-ada), proposed system without adaptation (DM) and the proposed system with adaptation (DM-ada). When applied without adaptation, the proposed method (DM) outperforms the baseline (Baseline) in all conditions. When each system is applied with adap-

**Table 1**. Evaluation tests on clean conditions. The results are shown for two baselines without (Baseline) and with (Baseline-ada) CMLLR-adaptation, and similarly two versions of the proposed method without (DM) and with CMLLR-adaptation (DM-ada) are shown. The best results are bolded.

| Method | Room | | | Ave. |
|---|---|---|---|---|
| | 1 | 2 | 3 | |
| Baseline | 12.89 | 12.64 | 12.13 | 12.55 |
| Baseline-ada | **11.78** | **11.42** | **11.21** | **11.47** |
| DM | 12.92 | 12.67 | 12.06 | 12.55 |
| DM-ada | 11.84 | 11.50 | 11.45 | 11.59 |

tation the proposed system (DM-ada) outperforms the adapted baseline (Baseline-ada) in all conditions. For *clean* data (Table 1), the best performing system is the baseline with CMLLR-adaptation (Baseline-ada) with a small margin compared to the second best (DM-ada).

Table 3 demonstrates effects of different parameters in the results in three issues using the development set data. First, the effect of temporal window length is addressed. If the performance is better for considerably greater window length than one, it can be used as a proof of the concept. The results that the system using $T = 20$ time windows outperforms one using window length $T = 1$ (no context). Second, we compare our primary approach with dimensionality reduction (40 component PCA) to the full dimensional PCA that retains all components, and observe that better results in all cases are obtained for 40 component PCA. Third, we compare results of our approach with and without the Wiener filter. We notice that without the Wiener filter, the performance drops even below that of the baseline system in several cases. When the parameter settings are contrasted we notice that having temporal context is more important than dimension of PCA (40 vs. full) and that the Wiener filter is important when the dimensionality reduction applied.

Figure 1 demonstrates computational time of the method without ASR back-end showing a real-time factor against length of the batch used for lookup table construction. The software implementation in Matlab is run in a single core of Intel(R) Xeon(R) CPU E3-1230 V2 @ 3.30GHz. The rightmost bar in the figure corresponds to the setting that was used to conduct ASR-simulations in this study. It corresponds to a full batch using development set Room 3 far-condition which is ca. 31 min in duration. The real-time factor in that case was 0.61. Required computation drops when batch length is reduced as less data is used for the lookup table construction. We did not pay particular attention to computation, thus with a little effort it should be possible to implement the approach with reduced computation. The present version involves unnecessary computation for the ease of implementation, such as reconstructing the lookup table for every utterance. These could be simply done once for each batch with a little effort in optimizing the implementation.

## 4. DISCUSSION

In this study we addressed speech dereverberation using an unsupervised single channel approach that utilizes speech prior, but takes only weak assumptions on reverberation. The assumptions that we take or that are built in our method are that long temporal context is required, reverberation is convolutive, and that we can successfully decorrelate both clean speech and reverberant spectral long context supervectors using PCA transformation learned for clean speech. Our results demonstrate that in the all reverberant cases we achieve better performance compared to the clean speech trained

**Table 2**. Evaluation tests on reverberant conditions. The results are shown for two baselines without (Baseline) and with (Baseline-ada) CMLLR-adaptation, and similarly two versions of the proposed method without (DM) and with CMLLR-adaptation (DM-ada) are shown. The best results are bolded.

| | SimData | | | | | | | RealData | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Room 1 | | Room 2 | | Room 3 | | Ave. | Room 1 | | Ave. |
| | Near | Far | Near | Far | Near | Far | – | Near | Far | – |
| Baseline | 18.32 | 25.77 | 42.71 | 82.71 | 53.56 | 87.97 | 51.82 | 90.07 | 88.01 | 89.04 |
| Baseline-ada | 14.86 | 19.10 | 24.59 | 64.48 | 34.16 | 79.34 | 39.40 | 82.88 | 80.49 | 81.68 |
| DM | 18.20 | 23.01 | 27.99 | 53.53 | 37.47 | 67.14 | 37.87 | 73.14 | 71.37 | 72.25 |
| DM-ada | **14.74** | **18.50** | **20.89** | **39.59** | **26.74** | **51.80** | **28.70** | **64.32** | **60.16** | **62.24** |

**Table 3**. Development test results on reverberant test conditions (all without CMLLR adaptation). Results are show baseline recognizer (Baseline), for three parameter settings of the proposed method, the version used for final tests utilizing $T = 20$ time frames (DM), the version over one time frame ($T = 1$), the version using $T = 20$ time frames and full dimensional PCA (full dim) and the version without Wiener filter (no filter). The best results are bolded.

| | SimData | | | | | | | RealData | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Room 1 | | Room 2 | | Room 3 | | Ave. | Room 1 | | Ave. |
| | Near | Far | Near | Far | Near | Far | – | Near | Far | – |
| Baseline | 15.29 | 25.29 | 43.90 | 85.80 | 51.95 | 88.90 | 51.81 | 88.71 | 88.31 | 88.51 |
| DM ($T = 20$) | **14.75** | **21.95** | **28.44** | **56.15** | **34.52** | **63.95** | **36.60** | **62.69** | **64.46** | **63.57** |
| $T = 1$ | 16.10 | 24.93 | 30.88 | 74.14 | 39.94 | 79.23 | 44.17 | 70.99 | 71.09 | 71.03 |
| full dim | 15.46 | 23.13 | 29.48 | 62.53 | 35.44 | 67.68 | 38.92 | 66.56 | 68.97 | 67.75 |
| no filter | 32.30 | 42.65 | 53.76 | 78.75 | 61.72 | 85.93 | 59.15 | 79.23 | 80.45 | 79.83 |

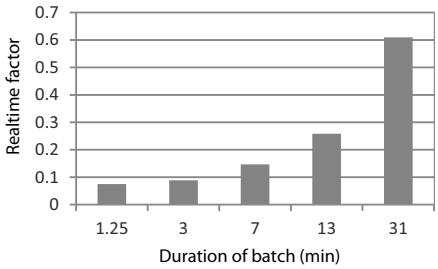


**Fig. 1**. Real-time factor as function of batch length used for distribution mapping.

baseline. Furthermore we showed that our method can be coupled to CMLLR adaptation with cumulative benefits.

Tests with different parameter settings on the system demonstrate it is essential to use long context in the supervector. In this paper we demonstrate results only comparing one frame temporal context to proposed 20 frame context, but our earlier development tests showed that 20 frame context was better than e.g. 10 frame context. Showing related results is omitted for compactness and as they are not compatible with the final version of the approach presented in this paper. Our results comparing the systems with and without Wiener filter demonstrate, first, that spectra originating directly from low dimensional PCA are overly smooth to represent speech accurately. Secondly, comparing the high dimensional PCA to the Wiener filtered low dimensional PCA, we notice that the Wiener filter is better at representing short term variation than utilizing the high dimensional PCA.

In the present study we chose to use full batch processing over utterance-wise from the challenge alternatives, because of need need to use adaptation data from reverberant posterior over one utterance duration. In development runs we conducted experiments with a version using a posterior model that is accumulated utterance by utterance, and the full batch version was only marginally better. Systematic investigations on the need of adaptation data is left, however, for future studies.

Regarding the computational load of the proposed method, it should be straightforward to implement it more efficiently. The first step would be to remove unnecessary computation that was left in the method for ease of implementation (see Sect. 3.3). Secondly, the computation could be reduced through histogram equalization using a more coarse distribution sampling [7]. After collecting sufficient data for reverberant posterior in the corresponding condition the method can applied with low latency of only one spectral frame if the supervector context is taken to represent the past mel-spectra. The question whether all full batch data is actually necessary is out of the scope of this study.

The present study used standard PCA to decorrelate spectral supervectors. During the development of the method we conducted also experiments with more sophisticated approaches such as stacked denoising auto-encoder (SDAE) [15] with which we generated bottleneck features using similar dimensionalities to PCA used here. With SDAE we obtained better speech reconstruction accuracies but coupling SDAE to the distribution mapping did not perform as well as the simpler PCA. Non-linear independent component analysis was also tried in earlier development stages. One of the reasons for the superior performance of PCA might be that we have learned mappings only from clean speech. Using data from posterior distribution in learning the mapping is certainly possible, but comes with increase in computation. However, using more sophisticated decorrelation methods is certainly among our main future interests.

## 5. REFERENCES

[1] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 29, pp. 254–272, 1981.

[2] B. E. D. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech Communication*, vol. 25, pp. 117–132, 1998.

[3] S. Thomas, S. Ganapathy, and H. Hermansky, "Recognition of reverberant speech using frequency domain linear prediction," *IEEE Signal Processing Letters*, vol. 15, pp. 681–684, 2008.

[4] K. J. Palomäki, G .J. Brown, and J.P. Barker, "Techniques for handling convolutional distortion with 'missing data' automatic speech recognition," *Speech Communication*, vol. 42, pp. 123–142, 2004.

[5] K. J. Palomäki, G. J. Brown, and J. Barker, "Recognition of reverberant speech using full cepstral features and spectral missing data," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP-2006)*, Tolouse, France, 2006, vol. 1, pp. 289–292.

[6] M.J.F. Gales, "Maximum likelihood linear transformations for hmm-based speech recognition," *Comput. Speech Lang.*, vol. 12, pp. 75–98, 1998.

[7] S. Dharanipragada and M. Padmanabhan, "A non-linear unsupervised adaptation technique for speech recognition," in *Proc. Int. Conf. Spoken Lang. Process. (ICSLP-2000)*, Beijing, 2000.

[8] A. Krueger, O. Walter, V. Leutnant, and R. Haeb-Umbach, "Bayesian Feature Enhancement for ASR of Noisy Reverberant Real-World Data," in *Proc. Interspeech*, Portland, USA, Sep. 2012.

[9] C. Evers, J. R. Hopgood, and J. Bell, "Blind speech dereverberation using batch and sequential monte carlo methods," in *IEEE Int. Symposium on Circuits and Systems*, May 2008, pp. 3226–3229.

[10] G. Saon, S. Dharanipragada, and D. Povey, "Feature space gaussianization," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP-2004)*, 2004, vol. I, pp. 329–332.

[11] J.F. Gemmeke, T. Virtanen, and A. Hurmalainen, "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Trans. Audio Speech Lang. Process.*, vol. 19, no. 7, pp. 2067–2080, 2011.

[12] K. Kinoshita, M. Delcroix, T. Yoshioka, T. Nakatani, E. Habets, R. Haeb-Umbach, V. Leutnant, A. Sehr, W. Kellermann, R. Maas, S. Gannot, and B. Raj, "The REVERB challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Proc. IEEE Workshop Applicat. Signal Process. Audio Acoust. (WASPAA-2013)*, 2013.

[13] T. Robinson, J. Fransen, D. Pye, J. Foote, and S. Renals, "WSJCAM0: a British English speech corpus for large vocabulary continuous speech recognition," in *Proc. Int. Conf. Acoust. Speech Signal Process. (ICASSP-1995)*, Detroit, MI, USA, 1995, pp. 81–84.

[14] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, "The HTK book, version 3.4," Tech. Rep., Cambridge University Engineering Department, 2006.

[15] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *J. Mach. Learn. Res.*, vol. 11, pp. 3371–3408, Dec. 2010.