REVISITING SOME MODEL-BASED AND DATA-DRIVEN DENOISING ALGORITHMS IN AURORA-2 CONTEXT

Panji Setiawan, Sorel Stan, Tim Fingscheidt

Siemens AG, ICM Mobile Phones
Grillparzerstr. 10-18, D – 81675 Munich, Germany

Abstract: In this paper we evaluate some model-based and data-driven algorithms for robust speech recognition in noise, using the experimental framework provided by ETSI Aurora 2. Specifically, we focus on statistical linear approximation (SLA), sequential interacting multiple models (S-IMM), and histogram normalization (HN). As the baseline for the feature extraction scheme we use the ETSI front-end.

Recognition tests on a subset of Aurora 2 show that SLA is approximately 4 % better than HN and that S-IMM is worse than HN by almost 3 % in terms of absolute word accuracy. A comparison with the ETSI advanced front-end (AFE) is also presented. While none of these algorithms outperforms AFE, we identify the reasons why this might have happened and point out potential directions for improvement.

1 Introduction

Recently, ETSI called for proposals for a noise robust speech processing front-end to be used in a distributed speech recognition set-up. In the scenario defined by ETSI STQ Aurora standardization body, the front-end resides in the mobile terminal and sends encoded speech vectors to the back-end running on a network server.

The winning denoising algorithm makes use of a two-stage Wiener filter, as well as a voice activity detector (VAD) for frame dropping. The processed speech vectors are converted back to the time domain and fed to a Mel filter cepstral coefficients (MFCC) front-end (FE), specified by [1]. The advanced front-end (AFE) [2] consists in essence of noise reduction, frame dropping, SNR-dependent waveform processing, and the standard Aurora FE.

This paper is motivated by the somehow surprising observation that at the core of the denoising algorithm in AFE is a classical speech enhancement technique, i.e. Wiener filtering, which challenges the belief that speech enhancement methods are suboptimal for automatic speech recognition (ASR) in noisy environments.

We revisit some denoising algorithms developed specifically for robust speech recognition and contrast them with AFE. We concentrate our attention on two model-based techniques, namely statistical linear approximation (SLA) [3] and sequential interacting multiple models (S-IMM) [4], as well as a data-driven method resulting from the experience we gained with two distinct histogram normalization (HN) algorithms [5], [6]. The two model-based techniques have also been investigated for speaker verification in noise [11]. All investigated algorithms are used in combination with the standard Aurora FE [1], which tested alone corresponds to “no denoising”. We identify the reasons why these algorithms fail to outperform AFE and point out potential improvements for model-based techniques.

The rest of the paper is organized as follows: section 2 describes the model-based algorithms, section 3 presents the data-driven techniques, section 4 describes the experiments, and finally section 5 summarizes our conclusions.

2 The Model-Based Approach

2.1 Noise Contamination Model

The effect of time domain additive noise on clean speech can be modeled in the log-spectral feature domain by a non-linear function [7]

$$\mathbf{z}_t = \mathbf{f}(\mathbf{x}_t, \mathbf{n_t}) = \mathbf{x}_t + \log[\mathbf{1} + \exp(\mathbf{n}_t - \mathbf{x}_t)], \tag{1}$$

where $\mathbf{x}_t$, $\mathbf{n}_t$, and $\mathbf{z}_t$ denote the clean speech, noise, and noisy speech log-spectral vectors, respectively.

We assume a Gaussian mixture model (GMM) with K components for the probability density function (*pdf*) of clean speech log-spectra and a single Gaussian model for the *pdf* of noise log-spectra

$$\begin{aligned} p(\mathbf{x}_t) &= \sum_{k=1}^{K} w_{\mathbf{x},k} \mathcal{N}(\mathbf{x}_t; \mu_{\mathbf{x},k}, \mathbf{\Sigma}_{\mathbf{x},k}) \\ p(\mathbf{n}_t) &= \mathcal{N}(\mathbf{n}_t; \mu_{\mathbf{n}}, \mathbf{\Sigma}_{\mathbf{n}}) \, . \end{aligned}$$

Given the environment model of Eq. 1, what is the distribution of the noisy speech vectors $\mathbf{z}_t$? If we replace $\mathbf{f}(\mathbf{x}_t, \mathbf{n_t})$ with the linear approximation

$$\mathbf{z}_t \approx \mathbf{A}_k \mathbf{x}_t + \mathbf{B}_k \mathbf{n}_t + \mathbf{c}_k, \quad k = \overline{1, K} \tag{2}$$

then each Gaussian of clean speech log-spectra will transform into a corresponding Gaussian of noisy speech log-spectra, so that the distribution of $\mathbf{z}_t$ is also a GMM with mean vectors and covariance matrices given by

$$\begin{aligned} \mu_{\mathbf{z},k} &= \hat{\mathbf{A}}_k \mu_{\mathbf{x},k} + \hat{\mathbf{B}}_k \mu_{\mathbf{n}} + \hat{\mathbf{c}}_k \\ \Sigma_{\mathbf{z},k} &= \hat{\mathbf{A}}_k \mathbf{\Sigma}_{\mathbf{x},k} \hat{\mathbf{A}}_k' + \hat{\mathbf{B}}_k \mathbf{\Sigma}_{\mathbf{n}} \hat{\mathbf{B}}_k' \, . \end{aligned} \tag{3}$$

In the equation above the prime denotes transposition.

2.2 Statistical Linear Approximation

SLA generalizes the vector Taylor series (VTS) algorithm introduced in [7], where the linear approximation is simply the first order Taylor polynomial expansion of $\mathbf{f}(\mathbf{x}_t, \mathbf{n_t})$. In contrast to VTS, SLA [3] computes the coefficients of Eq. 2 by minimizing the mean squared error between the environment function and its Taylor polynomial of m-th order. The coefficients are obtained as a function of the unknown noise statistics $\lambda_{\mathbf{n}} = \{\mu_{\mathbf{n}}, \mathbf{\Sigma}_{\mathbf{n}}\}$.

Similarly with VTS, an EM algorithm is then employed to improve iteratively the initial guess of the noise statistics and implicitly the linear approximation. Input to the EM algorithm is the model of clean speech log-spectra, given by the GMM parameter set $\lambda_{\mathbf{x}} = \{w_{\mathbf{x},k}, \mu_{\mathbf{x},k}, \mathbf{\Sigma}_{\mathbf{x},k}\}$, as well as the observed noisy speech log-spectra $\mathbf{Z} = \{\mathbf{z}_t\}_{t=\overline{1,T}}$.

The EM algorithm starts with an initial guess for the noise statistics $\hat{\lambda}_{\mathbf{n}} = \hat{\lambda}_{\mathbf{n}}^{(0)}$, computed from the first few non-speech frames, and improves it gradually to become $\hat{\lambda}_{\mathbf{n}}^{(i)}$ in the i-th step. The coefficients $\mathbf{A}_k, \mathbf{B}_k, \mathbf{c}_k$ are updated for each new estimate $\hat{\lambda}_{\mathbf{n}}^{(i)}$. The EM algorithm iterates until the likelihood function has converged. Using the final estimate of noise statistics and coefficients, the unobserved clean speech log-spectra are computed using the minimum mean squared error (MMSE) estimator

$$\hat{\mathbf{x}}_t = \mathbf{z}_t - \sum_{k=1}^{K} p(k \mid \mathbf{z}_t, \hat{\lambda}_{\mathbf{n}})(\mu_{\mathbf{z},k} - \mu_{\mathbf{x},k}) \,. \tag{4}$$

2.3 Sequential Interacting Multiple Model

While SLA provides only one estimate of noise statistics for the whole sequence $\mathbf{Z}$ (therefore implying stationarity), S-IMM provides an estimate $\hat{\lambda}_{\mathbf{n},t}$ of the noise in each log-spectral vector $\mathbf{z}_t$ [4]. The algorithm employs a bank of K Kalman filters, which share the state transition equation but have different observation models, i.e.

$$\begin{aligned} \mathbf{n}_t &= \mathbf{n}_{t-1} + \mathbf{u}_{t-1} \\ \mathbf{z}_t &= \hat{\mathbf{A}}_k \mathbf{x}_t + \hat{\mathbf{B}}_k \mathbf{n}_t + \hat{\mathbf{c}}_k \,. \end{aligned} \tag{5}$$

Note that the noise log-spectral vector is treated as the state of interest, and $\mathbf{u}_t$ is a zero-mean Gaussian process.

S-IMM uses an initial guess for $\hat{\lambda}_{\mathbf{n},t=1}$ and applies the Kalman prediction/update scheme to get an estimate of noise statistics for each mixture component. The K estimates are combined in the mixing step to obtain a single estimate

$$\begin{aligned} \hat{\mu}_{\mathbf{n},t} &= \sum_{k=1}^{K} p(k \mid \mathbf{Z}_t) \hat{\mu}_{\mathbf{n},t,k} \\ \hat{\mathbf{\Sigma}}_{\mathbf{n},t} &= \sum_{k=1}^{K} p(k \mid \mathbf{Z}_t) \left(\hat{\mathbf{\Sigma}}_{\mathbf{n},t,k} + \Delta\hat{\mu}_{\mathbf{n},t,k} \, \Delta\hat{\mu}'_{\mathbf{n},t,k} \right) \,, \end{aligned} \tag{6}$$

where $\mathbf{Z}_t = \{\mathbf{z}_1, \ldots, \mathbf{z}_t\}$ and $\Delta\hat{\mu}_{\mathbf{n},t,k} = (\hat{\mu}_{\mathbf{n},t,k} - \hat{\mu}_{\mathbf{n},t})$. This estimate is used to recompute $\hat{\mathbf{A}}_k, \hat{\mathbf{B}}_k, \hat{\mathbf{c}}_k$ for the next frame and as the initial value for $\hat{\lambda}_{\mathbf{n},t+1}$.

3 The Data-Driven Approach

3.1 Principle of Histogram Normalization

The goal of histogram normalization (HN) is to provide a transformation $T(\cdot)$ which maps each observed noisy speech vector to a "good" estimate of the unobserved clean speech vector. HN defines the sought after transformation as the mapping from the cumulative density function (*cdf*) of noisy speech to a reference *cdf* of clean speech [5], [6].

Let $\mathbf{z}$ represents noisy speech with *cdf* $C_{\mathbf{z}}(\mathbf{z})$ and $\hat{\mathbf{x}}$ be the estimated clean speech, whose *cdf* $C_{\hat{\mathbf{x}}}(\hat{\mathbf{x}})$ matches the reference *cdf* of clean speech $C_{\mathbf{x}}(\mathbf{x})$, i.e. $C_{\hat{\mathbf{x}}}(\hat{\mathbf{x}}) = C_{\mathbf{x}}(\mathbf{x})$. Then the transformation $T(\cdot)$ is defined as follows

$$\hat{\mathbf{x}} = T(\mathbf{z}) = \left(C_{\mathbf{x}}^{-1} \circ C_{\mathbf{z}} \right)(\mathbf{z}) = C_{\mathbf{x}}^{-1} \left[C_{\mathbf{z}}(\mathbf{z}) \right] \,. \tag{7}$$

In other words, HN replaces $\mathbf{z}$ by $\hat{\mathbf{x}}$ so that the equality $C_{\mathbf{z}}(\mathbf{z}) = C_{\mathbf{x}}(\hat{\mathbf{x}})$ holds.

3.2 Implementation of Histogram Normalization

The histogram bins can be uniformly distributed [6] over the input range $[\mathbf{z}_{\min}, \mathbf{z}_{\max}]$ of noisy speech. However, this method is suboptimal for short utterances as some bins may contain no data, therefore HN with non-uniform bins [5] provides the better alternative. The "optimal" non-uniform bins $[Q_{i-1}, Q_i)$ are selected such that $C_{\mathbf{z}}(Q_i) = i/N_Q$, where N_Q is the total number of bins. Note that this corresponds to dividing the range of input *cdf* in uniformly-spaced intervals.

In [6] not only the noisy speech (testing data) but also the clean speech (reference/training data) are transformed to a Gaussian distribution with zero mean and unit variance. Our implementation uses a combination of the methods presented in [5] and [6], in that we use non-uniform bins [5] and transform the clean speech reference data to a standard Gaussian [6] prior to processing of noisy speech.

Let $Q_i^{\mathbf{z}}$ and $Q_i^{\mathbf{x}}$ be the quantiles of the noisy and respectively clean speech data. For *cdf* matching we simply use linear interpolation, i.e.

$$\hat{\mathbf{x}} = C_{\mathbf{x}}^{-1}[C_{\mathbf{z}}(z)] \approx a_i \cdot \mathbf{z} + b_i \,, \tag{8}$$

where the coefficients are obtained for $\mathbf{z} \in \left[Q_{i-1}^{\mathbf{z}}, Q_i^{\mathbf{z}}\right)$ as

$$\begin{aligned} a_i &= \frac{Q_i^{\mathbf{x}} - Q_{i-1}^{\mathbf{x}}}{Q_i^{\mathbf{z}} - Q_{i-1}^{\mathbf{z}}} \\ b_i &= Q_{i-1}^{\mathbf{x}} - a_i \cdot Q_{i-1}^{\mathbf{z}} \,. \end{aligned} \tag{9}$$

Since multi-dimensional *cdf* matching is impracticable, HN is done using marginal 1-D histograms. This is equivalent to matching the multi-dimensional *cdf* only if the vector components are independent, which is clearly not the case.

4 Experimental Results

4.1 Database and Setup

All simulations were done on a subset of the ETSI Aurora 2 experimental framework, that is clean training set and test set A. The recognizer was trained using HTK as proposed by the ETSI STQ Aurora working group following [8]. The input to HTK are features with 39 coefficients consisting of 13 MFCCs plus delta and delta-delta coefficients. A single vector corresponds to a frame of length 25 ms and a frame shift of 10 ms.

For the case of SLA and S-IMM algorithms, the log energy was replaced by the zero-th cepstral coefficient $c(0)$. Compression and coding methods were not applied to any of the front-end schemes.

4.2 Performance Evaluation

In order to assess the relative merits of the algorithms under investigation, we rely on the word accuracy, which is the standard performance measure used by the ETSI STQ Aurora standardization body. The word accuracy is defined as

$$\text{Word Accuracy} = \frac{N - D - S - I}{N} \times 100\%, \tag{10}$$

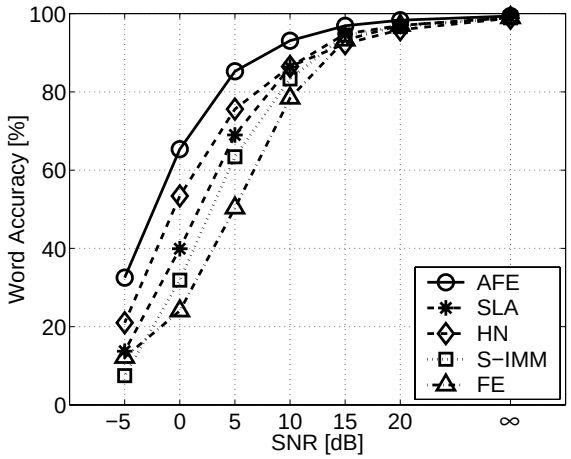

Figure 1 - Word accuracy on subway noise

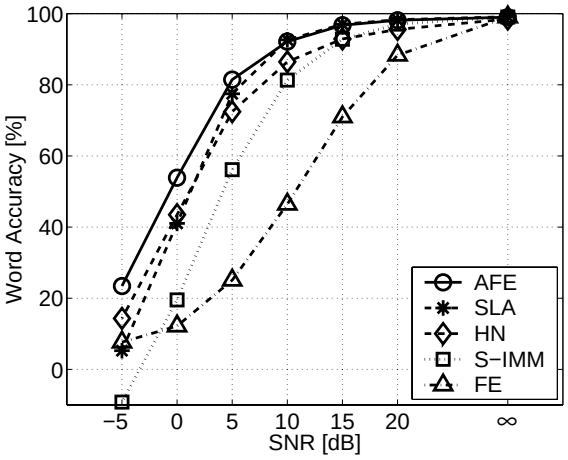

Figure 2 - Word accuracy on babble noise

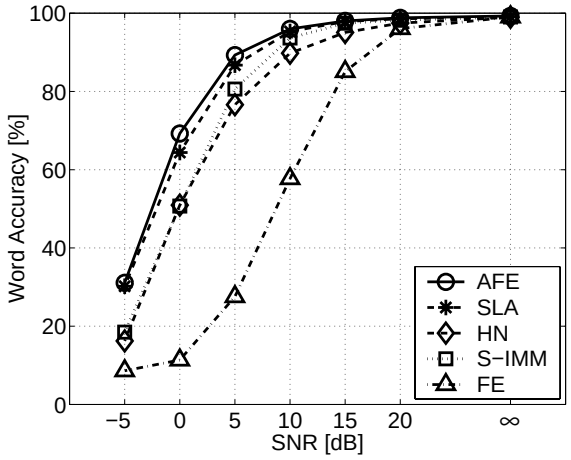

Figure 3 - Word accuracy on car noise

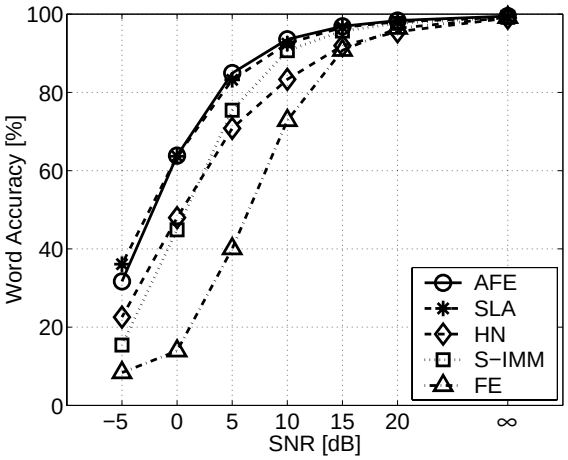

Figure 4 - Word accuracy on exhibition noise

where N is the total number of words in the reference labels, D denotes deletions, S substitutions and I insertions. Please note that the word accuracy can take on negative values if the number of insertions is large.

Another performance measure, i.e. word recognition rate, is also taken into consideration which is defined as

$$\text{Word Recognition Rate} = \frac{N - D - S}{N} \times 100\%. \tag{11}$$

4.3 Results and Analysis

Analysis in word accuracy on each type of noise shows that HN performs better than S-IMM in subway and babble noises as shown in Figures 1 and 2, and performs worse in car and exhibition noises as shown in Figures 3 and 4, especially at high SNR. The performance of SLA is very much comparable with that of AFE in car and exhibition noises.

In order to better visualize the overall performance, a global word accuracy was computed for

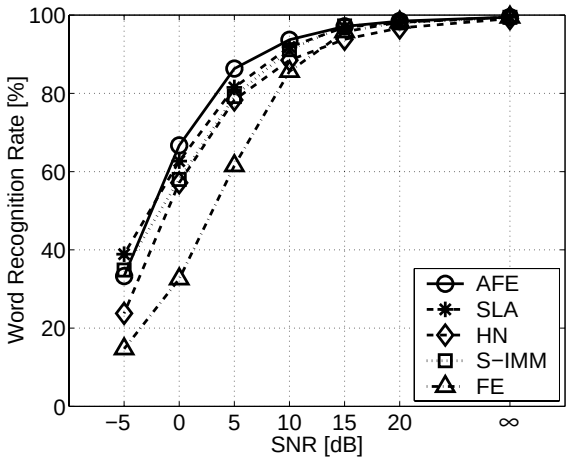

Figure 5 - Word recognition rate on subway noise

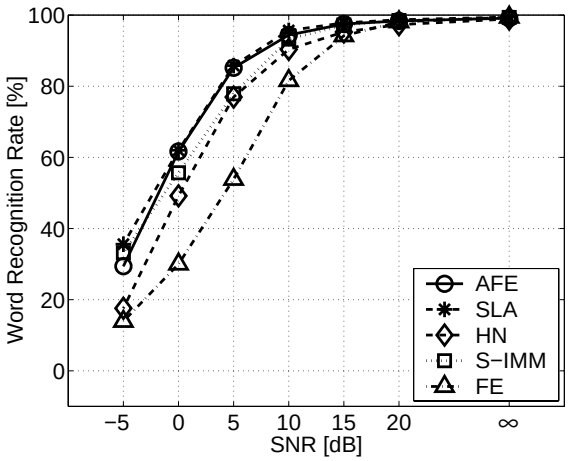

Figure 6 - Word recognition rate on babble noise

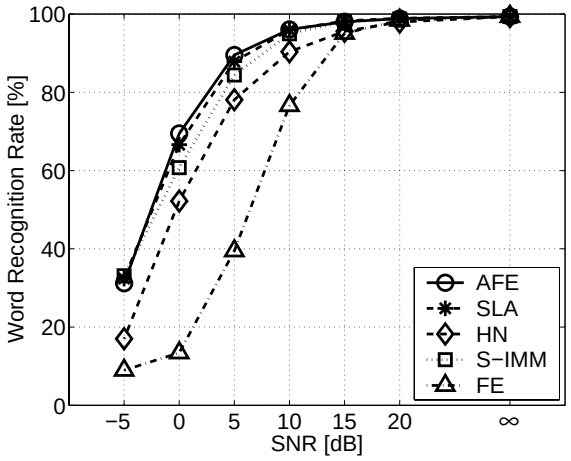

Figure 7 - Word recognition rate on car noise

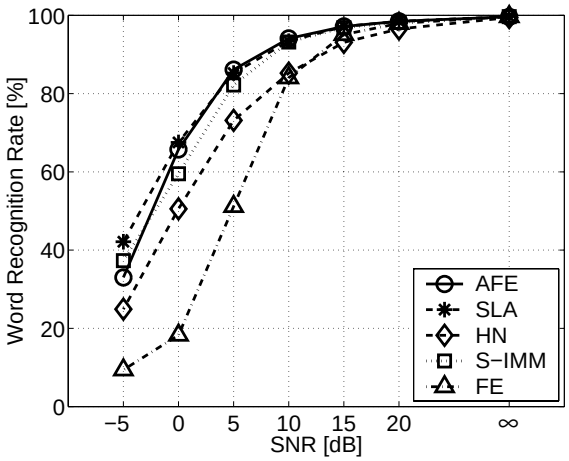

Figure 8 - Word recognition rate on exhibition noise

each method by averaging its performance from 0 to 20 dB and also for all types of noise conditions. The results are shown in Table 1. Note that the perfomance of the FE was evaluated using $c(0)$ instead of the log energy. A complete description of the FE and AFE performance in word accuracy on Aurora 2 is shown in [9] and [10] respectively.

	FE	S-IMM	HN	SLA	AFE
Word Accuracy [%]	58.89	77.08	79.70	83.53	87.51

Table 1 - Average word accuracy over all noises and SNRs.

Based on the results from Table 1, SLA as a model-based technique gives better results than HN, but this surprisingly does not hold for S-IMM. The AFE in this case performs really well compared to the other noise compensation methods. This could be explained by the fact that AFE employs not only a noise reduction scheme but also a feature vector selection (FVS) method. The FVS is basically a frame dropping method based on a VAD. It will improve the recognition

performance significantly by reducing the number of insertion errors which occur for speech preceeded and succeeded by long noisy non-speech segments.

Lower word accuracy for model-based noise compensation methods is attributed to the limitations of the environment model of Eq. 1 which holds only for speech segments. During non-speech segments, the model is not applicable, resulting in a high residual noise which causes a large number of insertion errors. This limitation does not exist in HN and particularly not in AFE, which drops the frames detected as silence. The word recognition rate is therefore useful to actually observe the potential of having a smaller amount of insertion errors. Figures 5, 6, 7 and 8 show the word recognition rate on subway, babble, car and exhibition noise respectively.

The performance of SLA as shown in Figures 5 and 6 is now comparable to the AFE on the other two noise conditions, i.e. subway and babble noise, where it was previously reported worse in term of word accuracy. S-IMM in this case performs better than HN in all noise conditions although it is still below SLA. An overall performance which employs similar calculation method as for word accuracy is obtained and shown in Table 2. It clearly supports the observation based on the figures stated earlier.

	FE	S-IMM	HN	SLA	AFE
Word Recognition Rate [%]	70.04	85.65	81.81	88.03	88.64

Table 2 - Average word recognition rate over all noises and SNRs.

5 Conclusions

In this paper we have investigated two model-based denoising algorithms, SLA and S-IMM, and a data-driven method, HN, in the context of Aurora 2 evaluation. While none of these algorithms outperforms AFE, we identify the reasons why this might have happened as well as point out potential improvements for SLA and S-IMM.

The model-based algorithms use an environment degradation model which breaks down during non-speech segments, leading to high residual noise and hence a large number of insertion errors. A direction for improvement is to combine them with silence frame dropping using a VAD, and also cepstral mean subtraction. Histogram normalization, on the other hand, suffers from the independence assumption of the vector components.

Surely, with the exception of S-IMM none of the other two methods could compete for a better AFE, given that they are batch processing techniques and the latency requirements on AFE are very strict. However, higher latencies are allowed for applications such as voice control in mobile phones, making batch processing algorithms a viable alternative to AFE, provided a better performance is achieved.

References

[1] ETSI standard document: Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Front-End Feature Extraction Algorithm; Compression Algorithms, ETSI ES 201 108 V1.1.2, April 2000.

[2] ETSI standard document: Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-End Feature Extraction Algorithm; Compression Algorithms, ETSI ES 202 050 V1.1.1, October 2002.

[3] Kim, N. S.: Statistical Linear Approximation for Environment Compensation. In: IEEE Signal Processing Letters, vol. 5, no. 1, pp. 8–10, 1998.

[4] Kim, N. S.: Feature Domain Compensation of Nonstationary Noise for Robust Speech Recognition. In: Speech Communication, vol. 37, pp. 231–248, 2002.

[5] Hilger, F. and Ney, H.: Quantile-Based Histogram Equalization for Noise Robust Speech Recognition. In: Proc. of Eurospeech, vol. 2, pp. 1135–1138, Denmark, September 2001.

[6] de la Torre, A., Segura, J. C., Benitez, C., Peinado, A. M. and Rubio, A. J.: Non-linear Transformations of the Feature Space for Robust Speech Recognition. In: Proc. of ICASSP, vol. 1, pp. 401–404, Orlando, FL, May 2002.

[7] Moreno, P. J., Raj, B. and Stern, R. M.: A Vector Taylor Series Approach for Environment Independent Speech Recognition. In: Proc. of ICASSP, Atlanta, GA, May 1996.

[8] Hirsch, H. G. and Pearce, D.: The AURORA Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions. In: Proc. of ICSLP, vol. 4, 29-32, Beijing, China, October 2000.

[9] Aurora document no. AU/411/02: Speech Recognition Performance Comparison between AMR Speech Coding and the DSR Front-End (ETSI ES 201 108), April 5, 2002.

[10] Aurora document no. AU/410/02: Speech Recognition Performance Comparison between AMR Speech Coding and the Advanced DSR Front-End (ETSI ES 202 050), March 25, 2002.

[11] Suhadi, Stan, S., Fingscheidt, T. and Beaugeant, C.:An Evaluation of VTS and IMM for Speaker Verification in Noise. In: Proc. of Eurospeech, Geneva, Switzerland, September 2003.