16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

INVENTORY BASED SPEECH DENOISING WITH HIDDEN MARKOV MODELS

Xiaoqiang Xiao[*], *Peng Lee*[*], *and Robert M. Nickel*[#]

Department of Electrical Engineering[*]
The Pennsylvania State University
University Park, PA 16802, USA
xxx106@psu.edu

Department of Electrical Engineering[#]
Bucknell University
Lewisburg, PA 17837, USA
robert.nickel@bucknell.edu

ABSTRACT

We are presenting a new speech waveform inventory based approach for the denoising of speech. The method combines an inventory style parametric description of speech signals with a statistical analysis of the underlying parameter space in clean and noisy conditions. Sufficient parameter statistics for successful denoising can be learned from around 40 minutes of (clean speech) training data. Shorter training sets are feasible, but may lead to quality reductions. Manual transcription of the training data is not required.

The proposed procedure is intended for applications in which speaker enrollment and noise enrollment are feasible. Such applications include vehicular speaker-phone communication systems and jet pilot communication systems. The proposed method compares very favorably to commonly used waveform based denoising methods in both objective and subjective speech quality assessments.

1. INTRODUCTION

One of the most hampering factors in speech signal processing today is the distortion of speech signals with additive noise. Human listeners are usually able to (psychoacoustically) reject even high levels of background noise. In contrast, mild levels of noise can interfere significantly in automatic speech recognition and speech coding [1].

The various modern approaches to denoising of speech are mostly *waveform filtering* based methods. Waveform filtering implies that only limited assumptions are made about the specific nature of the underlying signal (i.e. than that it is an acoustic waveform). The most prominent examples of waveform processing are the spectral subtraction method developed by Boll in 1979 [1], the Wiener filtering extensions proposed by McAulay and Malpass in 1980 [2] and Ephraim and Malah in 1984 [2]. Other examples include schemes that employ wavelets [3] and modifications of the iterative Wiener filter and the Kalman filter [4].

More recently, model based denoising methods have been proposed. In model based denoising a deterministic or stochastic parametric model for a speech signal (and its properties) is used instead of a general waveform model. A popular choice for a speech model in this context is the harmonic plus noise model (HNM) which was studied (amongst others) by Zavarehei, Vaseghi, and Yan [5]. Accurate modeling and estimation of speech and noise gains via hidden Markov models was proposed by Zhao and Kleijn [6]. Codebooks of linear predictive coefficients and their employment for speech denoising within a Maximum-likelihood framework was studied by Srinivasan, Samuelsson, and Kleijn [7]. Hendriks, Heusdens, and Jensen considered a minimum mean square error approach for denoising that relies on a combined stochastic and deterministic speech model [8].

The model based speech denoising method proposed in this paper is inspired by the increasing success of *inventory based speech synthesis systems* [9]. We are assuming that speaker enrollment and noise enrollment are feasible for the given denoising task. The speaker enrollment procedure provides us with training data that can be appropriately clustered and used as an inventory for a "clean" speech signal model. We are augmenting the inventory with a statistical analysis of the speech signal under clean and noisy conditions. The details of the proposed method are summarized in section 2. Experimental results and performance studies are provided in section 3.

2. METHODS

A block diagram of the proposed method is shown in figure 1. The denoising procedure is divided into two main tasks: (A) a system training task (dashed arrows in figure 1) and (B) the signal denoising task (solid arrows). The system training task is sub-divided into the following three main steps:

(1) The development of a (clustered) speech-waveform-unit *inventory* and an associated codebook of MFCC vectors obtained from *clean* speech frames (training data). We will refer to the MFCC codebook as the *clean codebook.* The clean codebook serves as a *state space* to the hidden Markov model (HMM) discussed in step 3.

(2) The development of a matching MFCC codebook obtained from *noisy* speech frames, referred to as the *noisy codebook.* The noisy codebook serves as the *observation space* of the HMM (see section 2.1).

(3) The estimation of the parameters of the HMM from the given training data (see section 2.2). The parameters include the state transition probabilities and the observation probabilities at each state.

The denoising task consists of two steps:

(1) Feature extraction from the incoming noisy speech signal and the estimation of an underlying clean speech *state sequence* based on the trained HMM.

(2) Denoising of speech via *unit selection* and *unit concatenation* based on the estimated state sequence (resynthesis).

Before we discuss the two main components of the proposed method in detail it is beneficial to first introduce some notation. At the *denoising* stage we assume that we observe a speech signal $s[n]$ uttered by the enrolled speaker and dis-

torted by zero mean additive noise $v[n]$:

$$x[n] = s[n] + v[n]. \quad (1)$$

At the *training* stage we use $\hat{s}[n]$ and $\hat{x}[n]$ to, similarly, denote the clean and noisy versions of the *enrollment data.* System training is done off-line from speaker-specific pre-recorded *clean* training data. The *noisy* training data is created by adding pre-recorded noise at the targeted signal-to-noise ratio (SNR) to the clean data. For simplicity we assume that all training records of *clean* speech are concatenated into one long training sequence $\hat{s}[n]$. Similarly, $\hat{x}[n]$ denotes the matching concatenation of *noisy* data.

Throughout the paper we make use of speech *units* or *frames.* We represent a unit as a vector of N successive samples of a signal:

$$\mathbf{s}_n = [\ s[n]\ s[n+1]\ \ldots\ s[n+N-1]\ ]^{\mathrm{T}}. \quad (2)$$

The amount of overlap between adjacent frames is controlled by a step size L. If i denotes a unit (or frame) index then the associated vector is written as $\mathbf{s}_{iL}$. Symbols $\mathbf{x}_n$, $\mathbf{v}_n$, $\hat{\mathbf{x}}_n$, and $\hat{\mathbf{s}}_n$ are defined analogously to equation (2).

We use $\mathbb{S}$ to denote our *speech-waveform-unit inventory.* Set $\mathbb{S}$ consists of all clean training data frames $\hat{\mathbf{s}}_n$ ($\forall n$, i.e. with a step size of one) with the exception of data frames that are entirely silent[1]. Furthermore, we use $V^2 = \mathrm{E}\{\mathbf{v}_n^{\mathrm{T}}\mathbf{v}_n\}$ to denote the variance of the noise.

Denoising is performed by finding a mapping $\mathbf{x}_{iL} \rightarrow \hat{\mathbf{s}}_{n(i)}$ that associates a specific inventory frame $\hat{\mathbf{s}}_{n(i)}$ to every observed noisy frame $\mathbf{x}_i$. Note that this mapping is generally not fixed, but time-variant and context dependent. The details are described in section 2.3.

The resulting denoised signal $\tilde{s}[n]$ is obtained by linearly cross-fading the overlapping parts of adjacent reconstructed frames. If we use the notation $[\hat{\mathbf{s}}_{n(i)}]_k$ to indicate the k^{th} element of vector $\hat{\mathbf{s}}_{n(i)}$ for $k = 0 \ldots N-1$ and $[\hat{\mathbf{s}}_{n(i)}]_k = 0$ for $k < 0$ and $k \geq N$ then

$$\tilde{s}[m] = \textstyle\sum_i w[m - iL] \cdot [\hat{\mathbf{s}}_{n(i)}]_{m-iL}, \quad (3)$$

in which sequence $w[m]$ denotes an appropriate trapezoid/ triangle-shaped window of length N.

2.1 Inventory Design

The goal of the *inventory design* stage is to group all inventory elements $\hat{\mathbf{s}}_n$ that belong to a similar *phonemic function*[2] into the same class. The purpose of the grouping is to be able to study the statistical properties of the group as a whole and then apply a resulting statistical description in the denoising process.

The inventory design is performed in two steps. First we group sets of adjacent frames via an *intra-phonemic* clustering method. Similarities between non-adjacent frames are then analyzed and considered in a second step via an *inter-phonemic* clustering method.

[1] We consider frames to be entirely silent if the total frame energy falls below a certain minimal level.

[2] We are using the term *phonemic function* in reference to a general, function carrying unit of a language. The group *may* or *may not* match with an actual *phoneme* defined for that language.

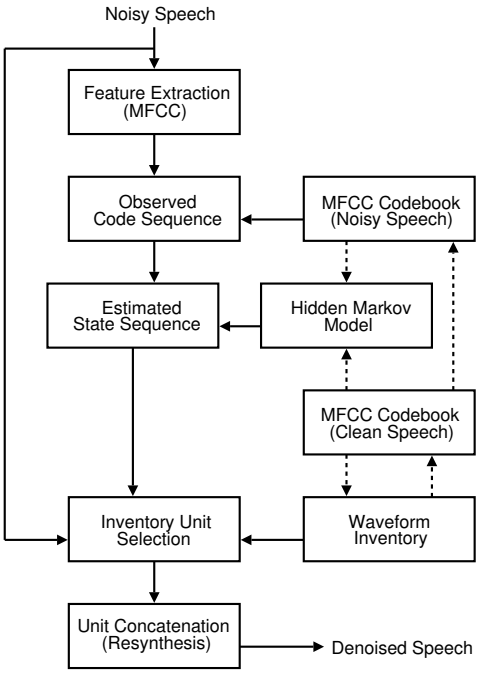

Figure 1. A block diagram of the proposed denoising method. Dashed lines indicate processing steps that are performed during system training.

2.1.1 Intra-Phonemic Clustering

To access intra-phonemic similarities between adjacent frames, we compute mel-frequency cepstral coefficients (MFCC) for each inventory frame $\hat{\mathbf{s}}_{iL}$.

$$\mathbf{c}_i = \mathrm{MFCC}\{\hat{\mathbf{s}}_{iL}\} \quad \text{with} \quad \dim(\mathbf{c}_i) = 12. \quad (4)$$

In our experiments (see section 3) we used a frame length N that corresponds to 20ms in length with a step size $L = N/2$. The employed MFCC computation procedure involves Hamming windowing in the time domain and triangular windowing in the *mel* domain (with windows that taper off to zero at zero frequency and at the Nyquist frequency) [10].

The intra-phonemic clustering procedure is described as follows. Assume that a successive set of frames $i \in \mathbb{I}_m$ was found to belong to a cluster set $\mathbb{I}_m$. We define the cluster centroid $\bar{\mathbf{c}}_{\mathbb{I}_m}$ with the simple averaging procedure:

$$\bar{\mathbf{c}}_{\mathbb{I}_m} = \tfrac{1}{\mathrm{size}\{\mathbb{I}_m\}} \textstyle\sum_{i \in \mathbb{I}_m} \mathbf{c}_i \quad (5)$$

The decision of whether to include the next incoming frame k into $\mathbb{I}_m$ is based on a threshold test:

$$\text{include } k \text{ in } \mathbb{I}_m \text{ if} \quad \|\bar{\mathbf{c}}_{\mathbb{I}_m} - \mathbf{c}_k\| < \lambda \quad (6)$$

$$\text{and} \quad \|\bar{\mathbf{c}}_{\mathbb{I}_m \cup \{k\}} - \mathbf{c}_i\| < \lambda \quad \text{for all} \quad i \in \mathbb{I}_m \quad (7)$$

If frame k fails to satisfy conditions (6) and (7) then frame k will be the first frame of the next index set $\mathbb{I}_{m+1}$. We continue to test for set membership to $\mathbb{I}_{m+1}$ for successive frames of k, and so forth.

The threshold value of λ determines the "phonemic granularity" of the approach. In our experiments we chose λ such that it would roughly represent one common phoneme unit. The resulting "phonemic granularity" is hence sufficiently fine to distinguish coarticulation effects. We found that $\lambda = 800$ worked best with the given training data (see section 3).

2.1.2 Inter-Phonemic Clustering

The purpose of inter-phonemic clustering is to study the similarity between non-adjacent frames. The difficulty in finding an appropriate clustering method is two-fold. We need to retain sufficient variety in less frequently occurring (more disconnected) frames to preserve intelligibility of the resulting denoised signal, while, at the same time, avoid large cluster sizes over densely populated areas to preserve a good overall reconstruction quality. In our experiments we chose a total number of $M = 50$ clusters to roughly match the variety of phonetic units in American English.

For inter-phonemic clustering we employ a commonly used two step approach: (1) we begin with a *k-center* clustering method (via a greedy algorithm [11]) followed by (2) a *k-means algorithm* [12]. The goal in the inter-phonemic clustering procedure is to reduce the large cluster number from the intra-phonetic clustering step by joining "similar" intra-phonetic clusters into bigger (and therefore less) inter-phonetic clusters. As an appropriate *cluster distance measure* we chose the Euclidean distance of the intra-phonemic clustering centroids:

$$\delta(m,p) = \|\bar{\mathbf{c}}_{\mathbb{I}_m} - \bar{\mathbf{c}}_{\mathbb{I}_p}\|. \tag{8}$$

We begin the clustering procedure with the two indices m_1 and m_2 that maximize the overall distance $\delta(m_1,m_2)$. We then use m_1 and m_2 as the seeds for two clusters $\mathbb{C}_{m_1}$ and $\mathbb{C}_{m_2}$ such that

$$i \in \mathbb{C}_{m_q} \quad \text{if} \quad \delta(i,m_q) = \min_{k=1,2} \delta(i,m_k). \tag{9}$$

We find the elements z_k for each cluster that are furthest removed from the respective cluster centers, i.e.

$$z_k = \arg\max_{i \in \mathbb{C}_{m_k}} \delta(i,m_k), \tag{10}$$

and define the seed for the next cluster $\mathbb{C}_{m_3}$ as

$$m_3 = \arg\max_{k=1,2} \delta(z_k,m_k). \tag{11}$$

We continue to iterate through equations (9) to (11) until we find the desired total number of clusters $\mathbb{C}_{m_1},\mathbb{C}_{m_2},\ldots,\mathbb{C}_{m_M}$ with the respective cluster seeds $m_1,m_2,\ldots,m_M$.

The *k-center* clustering procedure ensures that there are some initial cluster centroids near less frequently occurring frames. To reduce the size of excessively large clusters over highly populated feature space areas, the resulting cluster centroids are then used to initialize a *k-means algorithm* [12] to find a *new* set of clusters $\mathbb{C}_1,\mathbb{C}_2,\ldots,\mathbb{C}_M$. The resulting centroids $\mathscr{C}_k$ of all $\mathbf{c}_i$ for $i \in \mathbb{C}_k$ (for $k = 1\ldots M$) constitute our *clean* MFCC codebook:

$$\mathscr{C} = \{\mathscr{C}_1,\mathscr{C}_2,\ldots,\mathscr{C}_M\}. \tag{12}$$

We use the notation $\mathbb{K}_1,\mathbb{K}_2,\ldots,\mathbb{K}_M$ to denote the sets of all inventory segments $\hat{\mathbf{s}}_n$ that are mapped through their local affiliation sets $\mathbb{I}_m$ into one of the clusters $\mathbb{C}_k$. If segments $\hat{\mathbf{s}}_{iL}$ to $\hat{\mathbf{s}}_{jL}$ for $i,i+1,i+2,\ldots j-1,j$ belong to set $\mathbb{I}_m$ which is mapped into cluster $\mathbb{C}_k$ then all segments $\hat{\mathbf{s}}_n$ for $n = iL, iL+1, iL+2,\ldots,jL-1,jL$ are considered to be elements of $\mathbb{K}_k$.

Our *noisy* MFCC codebook $\widehat{\mathscr{C}}$ is also generated directly from the clusters $\mathbb{C}_k$ from above (for $k = 1\ldots M$). For every $i \in \mathbb{C}_k$ we generate an associate $\hat{\mathbf{x}}_{iL}$ by adding noise at the targeted SNR level to $\hat{\mathbf{s}}_{iL}$. We compute a resulting *noisy* MFCC vector

$$\hat{\mathbf{c}}_i = \text{MFCC}\{\hat{\mathbf{x}}_{iL}\} \quad \text{with} \quad \dim(\hat{\mathbf{c}}_i) = 12 \tag{13}$$

for every index $i \in \mathbb{C}_k$. The centroids $\widehat{\mathscr{C}}_k$ of the $\hat{\mathbf{c}}_i$ for all $i \in \mathbb{C}_k$ then form our *noisy* MFCC codebook:

$$\widehat{\mathscr{C}} = \{\widehat{\mathscr{C}}_1,\widehat{\mathscr{C}}_2,\ldots,\widehat{\mathscr{C}}_M\}. \tag{14}$$

The indices of the elements in our clean codebook $\mathscr{C}$ will serve as the *states* of the hidden Markov model that is discussed in the following section. The indices of the elements of our noisy codebook $\widehat{\mathscr{C}}$ will serve as the *observation codes* of the HMM.

2.2 HMM Parameter Estimation

To aid the denoising process we need to be able to judge the probability of a particular *state sequence* (i.e. the sequence of the "true" underlying MFCC representations) from a sequence of noisy MFCC vector observations (i.e. the codes from the noisy MFCC codebook). A probability measure for a particular state sequence can be computed if we have estimates of the *state transition probabilities* and the *state output probabilities* (i.e. the symbol/observation probabilities).

We apply vector quantization to convert an observed sequence of frames into an associated sequence of *states* or *observation codes* for clean and noisy speech respectively. The quantization is based on the following distortion measure

$$d(\mathbf{c}_r,\mathbf{c}_t) = \|\mathbf{c}_t\| \cdot \left(1 - \frac{\mathbf{c}_r^{\mathrm{T}}\mathbf{c}_t}{\|\mathbf{c}_r\|\|\mathbf{c}_t\|}\right), \tag{15}$$

which was found to be more robust under the considered noise conditions compared to Euclidean distances [13] (see section 3). Symbols $\mathbf{c}_r$ and $\mathbf{c}_t$ refer to the reference and the test MFCC vectors respectively. We define the quantization of a clean frame $\hat{\mathbf{s}}_n$ into *state k* via

$$\hat{\mathbf{s}}_n \to k \quad \text{if} \quad d(\mathscr{C}_k,\text{MFCC}\{\hat{\mathbf{s}}_n\}) = \min_{j=1,\ldots,M} d(\mathscr{C}_j,\text{MFCC}\{\hat{\mathbf{s}}_n\}). \tag{16}$$

Similarly, we define the quantization of a noisy frame $\hat{\mathbf{x}}_n = \hat{\mathbf{s}}_n + \mathbf{v}_n$ into *observation code k* via

$$\hat{\mathbf{x}}_n \to k \quad \text{if} \quad d(\widehat{\mathscr{C}}_k,\text{MFCC}\{\hat{\mathbf{x}}_n\}) = \min_{j=1,\ldots,M} d(\widehat{\mathscr{C}}_j,\text{MFCC}\{\hat{\mathbf{x}}_n\}). \tag{17}$$

By using these mappings we convert our sequence of training frames $\hat{\mathbf{s}}_{iL}$ into a *state sequence.* With a simple counting process we estimate the first-order temporal *state transition probabilities,* i.e.

$$P_{k,j} = \text{Prob}[\,\hat{\mathbf{s}}_{(i+1)L} \to j \,|\, \hat{\mathbf{s}}_{iL} \to k\,]. \tag{18}$$

Similarly, we convert our sequence of noisy training frames $\hat{\mathbf{x}}_{iL}$ into an *observation code sequence.* We estimate the noise induced *observation probabilities* jointly from our clean and noisy training data:

$$Q_{k,j} = \text{Prob}[\,\hat{\mathbf{x}}_{iL} \to j \,|\, \hat{\mathbf{s}}_{iL} \to k\,]. \tag{19}$$

The transition probabilities $P_{k,j}$ and $Q_{k,j}$ are both used in the proposed denoising process.

2.3 Speech Denoising

Speech denoising is performed on a frame-by-frame basis for each frame $\mathbf{x}_{iL}$ of the incoming noisy signal. In a first step we define a similarity measure between a noisy frame $\mathbf{x}_{iL}$ and an inventory element $\hat{\mathbf{s}}_n$:

$$\sigma(\mathbf{x}_{iL}, \hat{\mathbf{s}}_n) = \frac{\mathbf{x}_{iL}^{\mathrm{T}} \hat{\mathbf{s}}_n}{\sqrt{\|\mathbf{x}_{iL}\|^2 - V^2} \cdot \|\hat{\mathbf{s}}_n\|}. \quad (20)$$

The normalization of the correlation measure takes the estimated power $\sqrt{\|\mathbf{x}_{iL}\|^2 - V^2}$ of the clean speech $s[n]$ into account. For the estimate of the power of $s[n]$ we assume that the noise $v[n]$ is (approximately) orthogonal to $s[n]$.

For every noisy signal frame $\mathbf{x}_{iL}$ we extract its MFCC feature $\hat{\mathbf{c}}_i$ using equation (13). We can find the associated *observation code sequence* with equation (17). We then incorporate the trained HMM to find the "true," i.e. most probable, *state sequence.* The state sequence $k_{\mathrm{opt}}(i)$ that maximizes the observation probability is readily found via the Viterbi algorithm [1]. The resulting $k_{\mathrm{opt}}(i)$ determines the estimated cluster affiliation for every noisy frame $\mathbf{x}_{iL}$.

We proceed by finding the best intra-cluster match for every noisy signal segment $\mathbf{x}_{iL}$ in the affiliated cluster $\mathbb{K}_k$:

$$\hat{\mathbf{s}}^{(i,k)} = \underset{\hat{\mathbf{s}}_n \in \mathbb{K}_k}{\arg\max}\, \sigma(\mathbf{x}_{iL}, \hat{\mathbf{s}}_n). \quad (21)$$

Denoising is done by replacing each frame $\mathbf{x}_{iL}$ with the inventory frame $\hat{\mathbf{s}}^{(i,k_{opt}(i))}$:

$$\mathbf{x}_{iL} \quad \rightarrow \quad \hat{\mathbf{s}}^{(i,k_{opt}(i))}. \quad (22)$$

The denoised frames are then "stitched" back together by linearly cross-fading adjacent frames over the respective overlapping regions of $N - L$ samples as described by equation (3).

3. EXPERIMENTAL RESULTS

We evaluated the performance of the proposed methods with experiments over the `CMU_ARCTIC` database from the Language Technologies Institute at Carnegie Mellon University[3]. The database was generated specifically for the design of (inventory based) speech synthesis systems. The corpus subset that is used for our study stems from the *US English* male speaker with identifier `BDL`. It contains 1132 phonetically balanced English utterances, most of which are between one and four seconds long. The data is appropriately low-pass filtered and downsampled to a processing sampling rate of 8kHz.

Additive noise was taken from the `NOISEX` database from the Institute for Perception-TNO, The Netherlands Speech Research Unit, RSRE, UK[4]. For our experiments we used *white* noise and *buccaneer jet cockpit* noise.

In a first step we verified the validity of our HMM approach. We split the training data into two disjoint parts, estimated the *state transition probabilities* (equation (18)) and the *observation probabilities* (equation (19)) separately for each part and verified that the estimates were very similar across the two parts[5].

We proceeded by evaluating the performance of the proposed HMM supported search denoising scheme under additive *white* noise and *buccaneer jet cockpit* noise at a signal to noise ratio of 10 dB. From the database we chose 10 testing utterances and a randomly selected subset of 1000 training utterances. The training and testing sets were mutually disjoint and generally recorded in different sessions (morning vs. afternoon session). For the signal segmentation and inventory generation described in section 2 we used a frame length N of 160 samples and a step size L of 80 samples.The size M of the employed inventory was 50 clusters.

[3]The corpus is available at <http://www.festvox.org/cmu_arctic>.

[4]The noise is available at <http://spib.rice.edu/spib/select_noise.html>.

[5]The mean deviation between the two sets of probability was below 0.6%. The variance of the deviation was below 0.012%.

Table I
Average Objective Quality Measures for Conventional Enhancement Methods and the Proposed Method under Additive White Noise at 10dB SNR.

Quality Measures	Conventional Speech Enhancement			
	Noisy	Wiener Filter	MMSE STSA	CB-Driven Wiener Filter
PESQ	1.8	2.6	2.3	2.5
LLR	1.1	1.0	0.7	1.0
CEPD	5.6	5.8	4.5	5.7
Quality Measures	**Proposed Speech Enhancement**			
	Noisy	Known State Sequence	Plain Inventory Search	HMM Supported Search
PESQ	1.8	**2.9**	2.6	**2.7**
LLR	1.1	**0.5**	0.8	**0.6**
CEPD	5.6	**3.7**	4.7	**4.3**

Table II
Average Objective Quality Measures for Conventional Enhancement Methods and the Proposed Method under Additive Jet Cockpit Noise at 10dB SNR.

Quality Measures	Conventional Speech Enhancement			
	Noisy	Wiener Filter	MMSE STSA	CB-Driven Wiener Filter
PESQ	2.0	2.5	2.4	**2.6**
LLR	0.8	0.9	0.6	0.9
CEPD	4.7	5.1	**4.1**	4.8
Quality Measures	**Proposed Speech Enhancement**			
	Noisy	Known State Sequence	Plain Inventory Search	HMM Supported Search
PESQ	2.0	**2.7**	2.4	**2.6**
LLR	0.8	**0.5**	0.9	**0.6**
CEPD	4.7	**4.1**	5.1	4.4

An objective quality assessment was performed with the *Log Likelihood Ratio* (LLR), the *Cepstral Distance Measure* (CEPD), and a *Perceptual Evaluation of Speech Quality* (PESQ). The PESQ measure, an ITU recommendation developed by Rix *et. al.* [14], correlates very well with *subjective quality* of speech. Note that the LLR and the CEPD are distortion measures (i.e. smaller values are better) whereas the PESQ is a quality measure (i.e. a bigger value is better). All measures are comprehensively described in [14].

3.1 Evaluation of the Proposed Method

Besides the proposed "HMM supported search" scheme we also implemented two other inventory based systems for comparison: a scheme that utilizes a "known state sequence" and a scheme that ignores all state affiliations ("plain inventory search"). The "plain inventory search" was performed by considering only one cluster $\mathbb{K}_1$ which consisted of the entire inventory $\mathbb{S}$. Denoising was done directly through equations (21), (22), and (3). No statistical information was used

and denoising was based entirely on σ-similarity (equation (20)) between the noisy signal and the clean inventory.

The "known state sequence" scheme performed denoising according to the proposed method, except that we replaced the estimated state sequence $k_{\text{opt}}(i)$ in equation (22) with the *true* underlying state sequence derived from the clean testing signal. The "known state sequence" is, of course, typically not known *a priori.* We included the results, nevertheless, to demonstrate the potiential of the proposed method with improved state estimation schemes.

The lower parts of tables I and II summarize the results of the performed experiments. The performance of the proposed method falls consistently between the "plain inventory search" method and the "known state sequence" scheme. The improvements due to the HMM support in the search are *significant* for *both,* white noise and jet cockpit noise. The result confirms the value and validity of the assumed statistical model. Note, furthermore, that the HMM supported method reduces the computational complexity of the plain search dramatically due to the reduction of the search space.

3.2 Comparison to Other Methods

For comparison, we also implemented three filtering based denoising methods: (1) the classic *minimum mean-square error short-time spectral amplitude* estimator (MMSE STSA) by Ephraim and Malah [2], (2) the *iterative Wiener filtering* scheme described in [2] (with 3 iterations), and (3) a state-of-the-art *codebook-driven*[6] *Wiener filtering* scheme similar to the one described in [7]. The CB-driven filtering also required substantial training with speech and noise[7].

The results are listed in tables I and II. The distortion/quality measures of the best performing algorithm in each catergory are shown in a boldface font[8]. The proposed "HMM supported search" significantly outperforms all of the three conventional methods for white noise. For jet cockpit noise the proposed method performs at least as good as, if not better than the conventional methods, with the exception of the CEPD measure. However, even in the CEPD case we still improve upon the state-of-the-art method proposed in [7].

The improvement comes at the cost of an increased complexity. The complexity of the proposed method is dominated by the intra-cluster search (equation (21)) and grows at an order of $K \log K$ with K being the sample-number per cluster (equivalent to 48 seconds in average).

We also conducted informal subjective listening tests. We judged the performance of the proposed method quite favorable when compared to the conventional methods. A particularly good quality was produced by the "known state sequence" scheme, which encourages further study of the subject. Preliminary experiments with non-stationary noise also provided promising results but require further refinement.

4. CONCLUSIONS

We presented a new method for the denoising of speech. Our approach is based on an *inventory style* speech re-synthesis scheme that utilizes a statistical analysis of the underlying parameter space. The required statistical descriptions were obtained from noise enrollment and from speaker enrollment in clean conditions. With experiments we have shown that the proposed method performs very well in comparison to commonly used waveform based denoising methods.

[6] Abbreviated as *CB-driven.*

[7] We used the same codebook clustering method for the "HMM supported search" and the "CB-driven Wiener filtering" to make the comparison fair between the two approaches.

[8] The measures of the "known state sequence" scheme are also listed in boldface, but are not used for comparison.

REFERENCES

[1] J. R. Deller, J. G. Proakis, and J. H. Hansen, *Discrete-Time Processing of Speech Signals*, Macmillan, New York, 1993.

[2] P. C. Loizou, *Speech Enhancement, Theory and Practice*, CRC-Press, 2007.

[3] Y. Hu and P. C. Loizou, "Speech enhancement based on wavelet thresholding the multitaper spectrum," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 1, pp. 59–67, Jan. 2004.

[4] A. Mouchtaris, J. Van der Spiegel, P. Mueller, and P. Tsakalides, "A spectral conversion approach to single-channel speech enhancement," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1280–1193, May 2007.

[5] E. Zavarehei, S. Vaseghi, and Q. Yan, "Noisy speech enhancement using harmonic-noise model and codebook-based post-processing," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1194–1203, May 2007.

[6] D. Y. Zhao and W. B. Kleijn, "HMM-based gain modeling for enhancement of speech in noise," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 3, pp. 882–892, 2007.

[7] S. Srinivasan, J. Samuelsson, and W. B. Kleijn, "Codebook driven short-term predictor parameter estimation for speech enhancement," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 163–176, Jan. 2006.

[8] R. C. Hendriks, R. Heusdens, and J. Jensen, "An MMSE estimator for speech enhancement under a combined stochastic/deterministic speech model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 2, pp. 406–415, Feb. 2007.

[9] D. O'Shaughnessy, "Modern methods of speech synthesis," *IEEE Circuits and Systems Magazine*, vol. 7, no. 3, pp. 6–23, 2007.

[10] X. Wang, A. Acero, and H.-W. Hon, *Spoken Language Processing*, Prentice Hall, Upper Saddle River, New Jersey, 2001.

[11] Cormen, Leiserson, and Rivest, *Introduction to Algorithms*, McGraw-Hill, 1990.

[12] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Recognition*, Wiley-Interscience, 2001.

[13] D. Mansour and B. H. Juang, "A family of distortion measures based upon projection operation for robust speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 11, no. 37, pp. 1659–1671, Nov. 1989.

[14] Y. Hu and P. Loizou, "Evaluation of objective measures for speech enhancement," *Proceedings of INTERSPEECH-2006*, Sept. 2006.