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 (psycho-acoustically) 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. 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 \(v[n]\):

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

At the training stage we use \(\hat{s}[n]\) and \(\bar{x}[n]\) 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, \(\bar{x}[n]\) 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:

\[
s_n = [s[n], s[n+1], \ldots, s[n+N-1]]^T.
\] (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 \(s_{il}\). Symbols \(x_m, v_m, \hat{s}_m,\) and \(\hat{x}_m\) are defined analogously to equation (2).

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

Denoising is performed by finding a mapping \(x_{il} \rightarrow \hat{s}_{n(i)}\) that associates a specific inventory frame \(\hat{s}_{n(i)}\) to every observed noisy frame \(x_l\). 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 \(\hat{s}[n]\) is obtained by linearly cross-fading the overlapping parts of adjacent reconstructed frames. If we use the notation \(\hat{s}_{n(i)}^k\) to indicate the \(k^{th}\) element of vector \(\hat{s}_{n(i)}\) for \(k = 0 \ldots N - 1\) and \(\hat{s}_{n(i)}^0 = 0\) for \(k < 0\) and \(k \geq N\) then

\[
\hat{s}[m] = \sum w[m-iL] \cdot \hat{s}_{n(i)}^k, \quad (3)
\]

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 \(\hat{s}_m\) 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.](image-url)
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 inter-phonemic clustering centroids:

$$
\delta(m, p) = \|\hat{c}_m - \hat{c}_p\|. 
$$  
(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 $C_{m_1}$ and $C_{m_2}$ such that

$$
i \in C_{m_k} \text{ if } \delta(i, m_k) = \min_{k=1,2} \delta(i, m_k). \quad (9)
$$

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

$$z_k = \arg \max_{i \in C_{m_k}} \delta(i, m_k), \quad (10)$$

and define the seed for the next cluster $C_{m_3}$ as

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

We continue to iterate through equations (9) to (11) until we find the desired total number of clusters $C_{m_1}, C_{m_2}, \ldots, C_{m_M}$ 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 $C_1, C_2, \ldots, C_M$. The resulting centroids $\hat{c}_k^i$ of all $c_k$ for $i \in C_k$ (for $k = 1 \ldots M$) constitute our clean MFCC codebook:

$$\mathcal{C} = \{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_M\}. \quad (12)$$

We use the notation $k_1, k_2, \ldots, k_M$ to denote the sets of all inventory segments $\hat{s}_n$ that are mapped through their local affiliation sets $\hat{S}_m$ into one of the clusters $C_k$. If segments $\hat{s}_m$ to $\hat{s}_M$ for $i, i + 1, i + 2, \ldots, j$, $j$ belong to set $\hat{S}_m$ which is mapped into cluster $C_k$ then all segments $\hat{s}_n$ for $n = i, i + 1, i + 2, \ldots, j$, $j$, $\hat{S}_m$ are considered to be elements of $k$.

Our noisy MFCC codebook $\hat{\mathcal{C}}$ is also generated directly from the clusters $C_k$ from above (for $k = 1 \ldots M$). For every $i \in C_k$ we generate an associate $\hat{x}_L$ by adding noise at the targeted SNR level to $\hat{s}_L$. We compute a resulting noisy MFCC vector $\hat{c}_i = \text{MFCC}\{\hat{x}_L\}$ with $\text{dim} (\hat{c}_i) = 12$. (13)

for every index $i \in \mathcal{C}_k$. The centroids $\hat{c}_k^i$ of the $\hat{c}_i$ for all $i \in \mathcal{C}_k$ then form our noisy MFCC codebook:

$$\hat{\mathcal{C}} = \{\hat{c}_1, \hat{c}_2, \ldots, \hat{c}_M\}. \quad (14)$$

The indices of the elements in our clean codebook $\mathcal{C}$ 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 $\hat{\mathcal{C}}$ 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

$$d(c_r, c_t) = \|c_r\| \cdot (1 - \frac{c_r^T c_t}{\|c_r\| \|c_t\|}), \quad (15)$$

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

$$\hat{s}_n \rightarrow k \text{ if } \arg \min_{j=1, \ldots, M} d(\hat{C}_k, \text{MFCC}\{\hat{s}_n\}). \quad (16)$$

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

$$\hat{x}_n \rightarrow k \text{ if } \arg \min_{j=1, \ldots, M} d(\hat{\mathcal{C}}, \text{MFCC}\{\hat{x}_n\}). \quad (17)$$

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

$$P_{k,j} = \text{Prob}[\hat{s}_{i+1} \rightarrow j | \hat{s}_i \rightarrow k]. \quad (18)$$

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

$$Q_{k,j} = \text{Prob}[\hat{x}_L \rightarrow j | \hat{s}_L \rightarrow k]. \quad (19)$$

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 $x_L$ of the incoming noisy signal. In a first step we define a similarity measure between a noisy frame $x_L$ and an inventory element $s_n$:
\[
\sigma(\mathbf{x}_L, \hat{s}_n) = \frac{\mathbf{x}_L^T \hat{s}_n}{\sqrt{\|\mathbf{x}_L\|^2 - \|\hat{s}_n\|^2}}.
\]

(20)

The normalization of the correlation measure takes the estimated power \(\sqrt{\|\mathbf{x}_L\|^2 - \|\hat{s}_n\|^2}\) 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 \(\mathbf{x}_L\), we extract its MFCC feature \(\mathbf{c}_L\), 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 \(\hat{k}_{\text{opt}}(i)\) that maximizes the observation probability is readily found via the Viterbi algorithm [1]. The resulting \(\hat{k}_{\text{opt}}(i)\) determines the estimated cluster affiliation for every noisy frame \(\mathbf{x}_L\).

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

\[
\hat{s}^{(i,k)} = \arg \max_{\hat{s}_n \in \mathbb{K}_k} \sigma(\mathbf{x}_L, \hat{s}_n).
\]

(21)

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

\[
\mathbf{x}_L \rightarrow \hat{s}^{(i,k_{\text{opt}}(i))}.
\]

(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 BD1. 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 10dB. 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.

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 \(S\). Denoising was done directly through equations (21), (22), and (3). No statistical information was used.

<table>
<thead>
<tr>
<th>Quality Measures</th>
<th>Noisy</th>
<th>Wiener Filter</th>
<th>MMSE STSA</th>
<th>CB-Driven Wiener Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>PESQ</td>
<td>1.8</td>
<td>2.6</td>
<td>2.3</td>
<td>2.5</td>
</tr>
<tr>
<td>LLR</td>
<td>1.1</td>
<td>1.0</td>
<td>0.7</td>
<td>1.0</td>
</tr>
<tr>
<td>CEPD</td>
<td>5.6</td>
<td>5.8</td>
<td>4.5</td>
<td>5.7</td>
</tr>
</tbody>
</table>

Table I

<table>
<thead>
<tr>
<th>Quality Measures</th>
<th>Noisy</th>
<th>Known State Sequence</th>
<th>Plain Inventory Search</th>
<th>HMM Supported Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>PESQ</td>
<td>1.8</td>
<td>2.9</td>
<td>2.6</td>
<td>2.7</td>
</tr>
<tr>
<td>LLR</td>
<td>1.1</td>
<td>0.5</td>
<td>0.8</td>
<td>0.6</td>
</tr>
<tr>
<td>CEPD</td>
<td>5.6</td>
<td>3.7</td>
<td>4.7</td>
<td>4.3</td>
</tr>
</tbody>
</table>

Table II

<table>
<thead>
<tr>
<th>Quality Measures</th>
<th>Noisy</th>
<th>Known State Sequence</th>
<th>Plain Inventory Search</th>
<th>HMM Supported Search</th>
</tr>
</thead>
<tbody>
<tr>
<td>PESQ</td>
<td>2.0</td>
<td>2.7</td>
<td>2.4</td>
<td>2.6</td>
</tr>
<tr>
<td>LLR</td>
<td>0.8</td>
<td>0.5</td>
<td>0.9</td>
<td>0.6</td>
</tr>
<tr>
<td>CEPD</td>
<td>4.7</td>
<td>4.1</td>
<td>5.1</td>
<td>4.4</td>
</tr>
</tbody>
</table>

Table II

\(^3\)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%.
and denoising was based entirely on $\sigma$-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 $k_{opt}(i)$ 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 potential 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” and “known state sequence” scheme, the performance of the proposed method with improved state estimation schemes.

3.2 Comparison to Other Methods

For comparison, we also implemented three filtering based denoising methods: (1) the classical 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 Wiener filtering scheme similar to the one described in [7]. The CB-driven filtering also required substantial training with speech and noise.

The results are listed in tables I and II. The distortion/quality measures of the best performing algorithm in each category are shown in a boldface font. 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 two methods for white noise and jet cockpit noise. 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.

REFERENCES


