MEMORY AND COMPLEXITY REDUCTION FOR INVENTORY-STYLE SPEECH ENHANCEMENT SYSTEMS

Robert M. Nickel[#/*] *and Rainer Martin*[*]

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

Institut für Kommunikationsakustik[*]
Ruhr-Universität Bochum
D-44780 Bochum, Germany
rainer.martin@rub.de

ABSTRACT

In this paper we are presenting a method that provides a dramatic reduction in memory requirement and computational complexity for an *inventory-style* speech enhancement scheme with only a small impact on the perceptual quality of the output of the system. *Inventory-style* or *corpus-based* speech enhancement generally attempts to generate a clean speech signal from a noisy speech signal by first estimating the characteristics of the underlying clean signal and then recreating it via corpus-based speech synthesis. As such, inventory-based enhancement is very different from most traditional methods which are typically relying on adaptive filtering or spectral subtraction. The advantage of inventory-based enhancement is its (principal) ability to deliver a very natural sounding output. A significant drawback is its large memory requirement and its large computational complexity (in comparison to traditional techniques)[1]. The method proposed in this paper allows for a flexible reduction of the memory requirement as a function of the desired perceptual quality of the output. A data reduction by almost factor 10 is achievable with only minor losses in perceptual quality. Furthermore, a significant reduction of computational complexity is a possible choice in the implementation of the procedure.

1. INTRODUCTION

Traditional methods for speech enhancement are typically based on adaptive filtering and/or spectral subtraction. Recent examples for advanced enhancement schemes that follow the traditional paradigm are the *harmonic emphasis and adaptive comb filter* approach by Jin *et al.* [1] and the *particle filter* based approach by Laska *et al.* [2]. One of the principle problems of filtering and spectral subtraction based methods is that there is an inherent tradeoff between the achievable level of noise reduction and the inevitable level of signal distortion. Recent approaches that attempt to optimally balance the two competing constraints are (among others) the *distortion minimized speech enhancement* proposed by Jo *et al.* [3] and the post-processing technique for the *regeneration of over-attenuated components* by Ding *et al.* [4].

The motivation behind *inventory-* or *corpus-based* enhancement schemes is to devise an enhancement paradigm that has (in principle) the ability to produce a very natural sounding output without significant signal distortions. First successful implementations of such a paradigm were (among others) published by Xiao and Nickel [5] as well as Ming *et al.* [6] in 2010. Both schemes use a *corpus-based* speech resynthesis approach to generate a "clean" speech output signal. The approach by Ming *et al.* employs a speech unit concatenation scheme with a flexible unit length whereas the approach by Xiao and Nickel employs a concatenation scheme based on a fixed unit length.

Inventory style speech enhancement systems have two significant disadvantages: 1) they typically require a very large amount of memory to store the inventory and 2) the computational complexity that is required to find the inventory unit that best matches an incoming noisy unit is typically very high. In this paper we are presenting a method for a significant reduction of the memory requirement as well as the computational complexity of such an approach. As our reference baseline we employ the procedure proposed by Xiao and Nickel in 2010 [5]. A block diagram of the baseline system, including the blocks that are added/modified in this paper, are shown in figure 1.

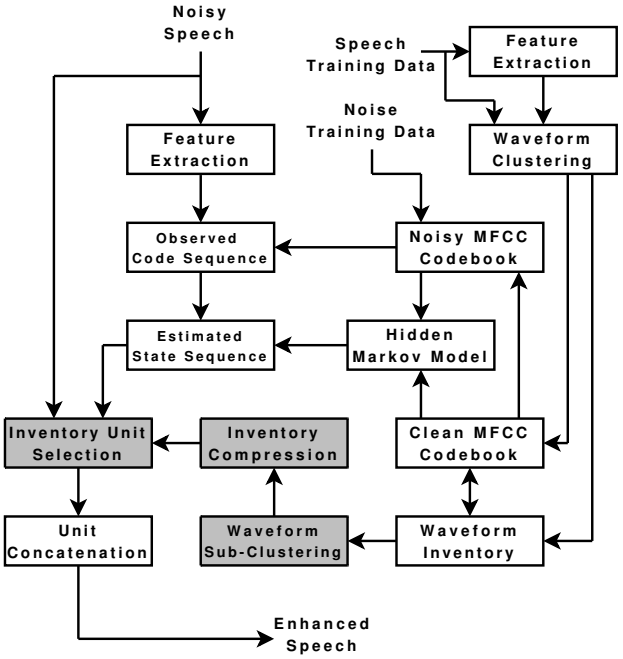

Figure 1. A block diagram of the proposed speech enhancement system. The method employs the inventory based scheme that was introduced by Xiao and Nickel in [5]. The sub-blocks that are added/modified in this paper are shaded in gray.

[1] Another significant drawback, specifically of the inventory-based enhancement method proposed in [5], is that it is speaker dependent with a processing latency of around 40 msec. Speaker dependency and latency, however, are issues that are not addressed in this paper.

It is beyond the scope of this paper to discuss the function of the entire baseline system in detail. The interested reader may consult reference [5] for a comprehensive description. For the purpose of this work it suffices to know that the part of the problem that causes the largest amount of computational complexity as well as the largest amount of required storage is a correlation procedure that attempts to find the best matching signal segment $s[n]$ for an incoming noisy segment $x[n]$ within a prescribed subset $\mathbb{S}$ of the inventory. Which subset is chosen for the search is decided in the baseline system by a *hidden Markov model* (see [5]).

2. METHODS

We assume that we have access to discrete time acoustic signals that were sampled at 16 kHz with a fine quantization granularity. We will use vectors to denote segments of these discrete time signals. For example, the vector

$$\mathbf{x} = [\quad x[n+1] \quad x[n+2] \quad \ldots \quad x[n+N] \quad]^{\mathrm{T}} \tag{1}$$

denotes a segment of length N from signal $x[n]$. Please note that the time alignment of the segment $\mathbf{x}$ at time n is, for simplicity, not explicitly made clear in our notation. The time alignment, however, will become clear in the context in which vector $\mathbf{x}$ will be used.

Throughout our presentation we employ a normalized correlation measure between two signal vectors $\mathbf{x}$ and $\mathbf{y}$:

$$\mathrm{corr}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}^{\mathrm{T}}\mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}. \tag{2}$$

We will also employ a specific notation to access subsegments of signal vector $\mathbf{x}$ via

$$\mathbf{subv}(\mathbf{x},p,q) = [\ x[n+p]\ x[n+p+1]\ \ldots\ x[n+q]\ ]^{\mathrm{T}}. \tag{3}$$

Throughout our discussion of the data reduction procedure we will refer to the speech inventory that is available to us as $s[n]$. We assume that the inventory has been divided into sub-sections $\mathbf{s}_i(m)$ of varying lengths $N_i(m)$. All subsections have been sorted according to a certain similarity condition (see [5] for the details) into one of 50 clusters $\mathbb{S}(m) = \{\ \mathbf{s}_1(m),\ \mathbf{s}_2(m),\ \mathbf{s}_3(m),\ \ldots\ \}$ with

$$\mathbf{s}_i(m) = [\ s[n_i(m)+1]\ s[n_i(m)+2]\ \ldots \\ \ldots\ s[n_i(m)+N_i(m)]\ ]^{\mathrm{T}}. \tag{4}$$

We use index m to indicate the cluster membership and index i to denote the segment index within cluster $\mathbb{S}(m)$.

The goal of the procedure that is presented in this paper is twofold: (1) we are aiming to reduce the storage requirement for the inventory, i.e. for the $\mathbb{S}(m)$ for $m = 1 \ldots 50$, and (2), at the same time, we are aiming to dramatically reduce the number of operations that is required to find a suitable inventory segment[2] $\mathbf{s} = [\ s[n^*+1]\ s[n^*+2]\ \ldots\ s[n^*+N]\ ]^{\mathrm{T}}$ for every incoming noisy segment $\mathbf{x}$ within a given cluster $\mathbb{S}(m)$.

2.1 Cluster Preprocessing

Within this section we will, for notational convenience, omit the explicit dependency on cluster index m, i.e. $\mathbb{S} = \{\ \mathbf{s}_1,\ \mathbf{s}_2,\ \mathbf{s}_3,\ \ldots\ \}$. The goal of the cluster preprocessing is to reorganize the unaligned, flexible length segments within a cluster into a set of time-aligned vectors of fixed length[3] L. We are considering the elements of cluster $\mathbb{S}$ to be elements of a "queue". The queue is subjected to an iterative procedure during which, at each iteration k, elements are added to the queue as well as removed from the queue. We will use the superscript k to indicate the queue at iteration k, i.e. $\mathbb{S}^k = \{\ \mathbf{s}_1^k,\ \mathbf{s}_2^k,\ \mathbf{s}_3^k,\ \ldots\ \}$. The queue is initialized with $\mathbb{S}^0 = \mathbb{S}$ such that $\mathbf{s}_i^0 = \mathbf{s}_i$ for all i.

We are furthermore initializing our collection $\mathbf{a}_k$ of time-aligned inventory vectors by choosing *that* inventory segment of $\mathbb{S}$ that lies symmetrically around the sample $s[n_{\max}]$ with the largest absolute value in $\mathbb{S}$, i.e.

$$\mathbf{a}_0 = [\ s[n_{\max}-L_{\mathrm{s}}]\ \ldots\ s[n_{\max}]\ \ldots\ s[n_{\max}+L_{\mathrm{e}}]\ ]^{\mathrm{T}} \tag{5}$$

with $L_{\mathrm{s}} = \mathrm{floor}(L/2)$ and $L_{\mathrm{e}} = L - L_{\mathrm{s}} - 1$. We also initialize an "averaged" version of $\mathbf{a}_k$ with $\bar{\mathbf{a}}_0 = \mathbf{a}_0$. For successive values of k (with initial value $k = 0$) we run through the following iteration:

1. We find the time index p^* and the queue index i^* of the inventory segment that best matches the current averaged segment $\bar{\mathbf{a}}_k$:
$$(i^*,p^*) = \underset{i,p}{\mathrm{argmax}} \left| \mathrm{corr}(\bar{\mathbf{a}}_k, \mathbf{subv}(\mathbf{s}_i^k, p, p+L-1)) \right| \tag{6}$$
2. We extract the matching segment and chose it (equipped with the proper sign) as our next time aligned vector:
$$\mathbf{a}_{k+1} = \mathbf{subv}(\mathbf{s}_{i^*}^k, p^*, p^*+L-1) \cdot \ldots \\ \mathrm{sign}(\mathrm{corr}(\bar{\mathbf{a}}_k, \mathbf{subv}(\mathbf{s}_{i^*}^k, p^*, p^*+L-1))) \tag{7}$$
3. We update our alignment reference vector $\bar{\mathbf{a}}_{k+1}$ with a learning rate of $\mu = 0.2$:
$$\bar{\mathbf{a}}_{k+1} = (1-\mu)\cdot\bar{\mathbf{a}}_k + \mu \cdot \mathbf{a}_{k+1} \tag{8}$$
The learning rate forces new $\mathbf{a}_k$ vectors to be as similar as possible to a collection of previous $\mathbf{a}_k$'s, yet allows the possibility, as k progresses, to slowly change and adapt to the variety of waveforms that are present in the given cluster.
4. We update our queue with
$$\mathbb{S}^{k+1} = \mathbb{S}^k - \{\mathbf{s}_{i^*}^k\} + \hat{\mathbb{S}}_R^{k+1} + \hat{\mathbb{S}}_L^{k+1} \tag{9}$$
in which $(-)$ refers to a removal from the queue an $(+)$ refers to an addition to the queue. Sets $\hat{\mathbb{S}}_L^{k+1}$ and $\hat{\mathbb{S}}_R^{k+1}$ are defined by
$$\hat{\mathbb{S}}_L^{k+1} = \begin{cases} \{\mathbf{subv}(\mathbf{s}_{i^*}^k, 1, p^*+M)\} & \text{if } M+p^* >= L \\ \{\ \} & \text{otherwise.} \end{cases} \tag{10}$$
$$\hat{\mathbb{S}}_R^{k+1} = \begin{cases} \{\mathbf{subv}(\mathbf{s}_{i^*}^k, p^*-M, L_{i^*}^k)\} & \text{if } \hat{M}-p^* >= L \\ \{\ \} & \text{otherwise.} \end{cases} \tag{11}$$
with $L_i^k = \mathrm{length}(\mathbf{s}_i^k)$ and $\hat{M} = L_{i^*}^k + M + 1$ and in which $\{\ \}$ denotes the empty queue.
5. We repeat the procedure until $\mathbb{S}^{k+1}$ is empty.

As a result of this iterative procedure and a subsequent normalization we obtain a sequence of time-aligned normalized inventory vectors $\{\ \tilde{\mathbf{a}}_1,\ \tilde{\mathbf{a}}_2,\ \tilde{\mathbf{a}}_3,\ \ldots\ ,\ \tilde{\mathbf{a}}_K\ \}$ in which $\tilde{\mathbf{a}}_k = \mathbf{a}_k / \|\mathbf{a}_k\|$.

[2] In our experiments from section 3 we used $N = 160$.

[3] In our experiments from section 3 we used $L = 353$ so that $L+N-1$ is a power of two value for faster processing.

2.2 Generation of Sub-Clusters

We are dividing our time-aligned clusters from section 2.1 furthermore into sub-clusters. As a first step we need to calculate the appropriate *number* of sub-clusters for each main cluster. It would not be appropriate to pick a fixed number of sub-clusters since the number of elements K in each cluster as well as the variety of waveforms within each cluster is significantly different. Finding the *optimal* number[4] of sub-clusters is, unfortunately, computationally prohibitive. We are therefore proceeding with a heuristic approach.

A direct measure B of waveform variation within a given main cluster is defined with

$$B = \sum_{k=1}^{K} \|\tilde{\mathbf{a}}_{k+1} - \tilde{\mathbf{a}}_k\| . \tag{12}$$

We use B_m to indicate the cluster dependency of the variation measure and, similarly, K_m to denote the number of vectors $\mathbf{a}_k$ in cluster m. It can be observed that B_m and K_m are strongly correlated (almost proportional), yet the "proportionality constant" between B_m and K_m is different for voiced and for unvoiced clusters.

We used a normalized version[5] of the average pre-emphasis energy ratio P_m that is described in [7] to measure the "voicing level" of each cluster. If we denote the "proportionality constants" of voiced and unvoiced clusters with α and β, respectively, then we can estimate the expected variation measure $\hat{B}_m$ for each cluster with

$$\hat{B}_m(\alpha,\beta) = \alpha \cdot (1 - P_m) \cdot K_m + \beta \cdot P_m \cdot K_m. \tag{13}$$

Vice versa, we can also use the *observed* variation measure B_m to estimate values for α and β in a least squares sense:

$$(\alpha^*, \beta^*) = \underset{\alpha,\beta}{\operatorname{argmax}} \left| B_m - \hat{B}_m(\alpha,\beta) \right|^2 . \tag{14}$$

If we assume that $V_{\text{voiced}} = 2L$ is the average number of $\mathbf{a}_k$'s for voiced sub-clusters and that $V_{\text{unvoiced}} = L$ is the average number of $\mathbf{a}_k$'s for unvoiced sub-clusters then the resulting number of average vectors per sub-cluster V_m for the given main cluster m computes as

$$V_m = V_{\text{voiced}} - \frac{B_m - \alpha^* \cdot K_m}{(\beta^* - \alpha^*) \cdot K_m} \cdot (V_{\text{voiced}} - V_{\text{unvoiced}}). \tag{15}$$

The estimated number Q_m of sub-clusters for main cluster m becomes $Q_m = \text{ceil}(\frac{K_m}{V_m})$. Furthermore, to protect against Q_m's that are too small, we are limiting each Q_m to a value larger or equal to 5.

As a last step we can now employ a (Euclidean distance) k-means algorithm with Q_m centroids on the set $\{\, \tilde{\mathbf{a}}_1^m, \tilde{\mathbf{a}}_2^m, \ldots, \tilde{\mathbf{a}}_{K_m}^m \,\}$ for each cluster $m = 1 \ldots 50$. The k-means algorithm provides a mapping $k_1(j)$, $k_2(j)$, ... (and so forth) that distributes the vectors $\tilde{\mathbf{a}}_k^m$ into Q_m matrices $\mathbf{A}_j^m$ such that $\mathbf{A}_j^m = [\, \tilde{\mathbf{a}}_{k_1(j)}^m \; \tilde{\mathbf{a}}_{k_2(j)}^m \; \tilde{\mathbf{a}}_{k_3(j)}^m \; \ldots \,]$ for $j = 1 \ldots Q_m$.

2.3 Data Compression

The reduction of the memory requirement for the inventory is accomplished with a *singular value decomposition* of each matrix $\mathbf{A}_j^m$:

$$\mathbf{U}_j^m \cdot \Sigma_j^m \cdot (\mathbf{V}_j^m)^{\mathrm{T}} = \mathbf{A}_j^m. \tag{16}$$

We assume that the singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_L$ are arranged on the diagonal of matrix Σ_j in descending order. We use $\mathbf{u}_k$ to denote the column vectors of matrix $\mathbf{U}_j^m$, i.e. $\mathbf{U}_j^m = [\, \mathbf{u}_1 \; \mathbf{u}_2 \; \ldots \; \mathbf{u}_L \,]$. Let's assume that there exists an index $\ell < L$ to construct an appropriately compressed *eigenvector matrix* $\tilde{\mathbf{U}}_j^m = [\, \mathbf{u}_1 \; \mathbf{u}_2 \; \ldots \; \mathbf{u}_\ell \,]$ and the associated *expansion coefficient matrix* $\tilde{\mathbf{\Phi}}_j^m = (\tilde{\mathbf{U}}_j^m)^{\mathrm{T}} \cdot \mathbf{A}_j^m$. A suitable approximation $\tilde{\mathbf{A}}_j^m$ to $\mathbf{A}_j^m$ is then given by

$$\tilde{\mathbf{A}}_j^m = \tilde{\mathbf{U}}_j^m \cdot \tilde{\mathbf{\Phi}}_j^m. \tag{17}$$

The quality of the resulting *compressed* inventory $\tilde{\mathbf{A}}_j^m$ can be measured with the *mean-squared-error* γ of the reconstruction:

$$\gamma(m,j,\ell) = \frac{\|\mathbf{A}_j^m - \tilde{\mathbf{A}}_j^m\|^2}{\|\mathbf{A}_j^m\|^2} = \frac{1}{c(m,j)} \sum_{k=\ell+1}^{L} \sigma_k^2, \tag{18}$$

in which we use $c(m,j)$ to denote the number of columns of matrix $\mathbf{A}_j^m$. Due to the (unit energy) column normalization of matrix $\mathbf{A}_j^m$ we obtain the *equivalent inventory reconstruction SNR* (EIR SNR) as

$$\text{EIR SNR}(m,j,\ell) = -10 \cdot \log_{10}(\gamma(m,j,\ell)). \tag{19}$$

For a given pre-defined EIR SNR_{def} we can select an optimal $\ell^*(m,j)$ for each m and j so that $\ell^*(m,j)$ is the smallest ℓ such that EIR $\text{SNR}(m,j,\ell) \geq$ EIR SNR_{def}. The number of coefficients $\mathcal{N}_c$ that need to be stored in memory for the resulting ℓ^*'s becomes

$$\mathcal{N}_c = \sum_{m,j} \min\{\, \ell^*(m,j) \cdot (L + c(m,j)) \,,\, L \cdot c(m,j) \,\}, \tag{20}$$

in which we recognize that for sub-clusters for which $\ell^*(m,j)$ is too big, it is more efficient to store $\mathbf{A}_j^m$ directly instead of $\tilde{\mathbf{U}}_j^m$ and $\tilde{\mathbf{\Phi}}_j^m$ separately.

2.4 Correlation Search and Complexity Reduction

The last aspect of our modification of the approach proposed in [5] pertains to the correlation search that needs to be performed for each incoming noisy signal segment $\mathbf{x}$ within a given cluster[6] m. If we want to perform an optimal (i.e. full) search within cluster m then we can find the best correlation match for each incoming segment $\mathbf{x}$ with each column of matrix $\tilde{\mathbf{A}}_j^m$ for all j in a procedure similar to equations (6) and (7). The correlation can be computed particularly fast if L+N-1 is a power of two number (see section 3.2).

A significant reduction of the computation complexity is possible if, instead of employing an exhaustive *full* search of each cluster, we employ a suboptimal *hierarchical* subsearch within each cluster. We can gage the likelihood of a good match for a given $\mathbf{x}$ within sub-cluster j by performing a correlation search across the first column of $\tilde{\mathbf{U}}_j^m$ only (i.e. with the dominant eigenvector). The j with the best eigenvector match is selected and a full search is performed across sub-cluster j only. Note, however, that for sub-clusters for

[4] Optimal in the sense of a maximum average *equivalent inventory reconstruction SNR* (EIR SNR) for a given compression rate, see section 2.3.

[5] Normalized such that $\max_m\{P_m\} = 1$. We have $0 \leq P_m \leq 1$.

[6] Again, the procedure that determines which cluster is chosen cannot be described here due to space limitations. Please refer to [5] for the details.

which we do not store $\tilde{\mathbf{U}}_j^m$ and $\tilde{\mathbf{\Phi}}_j^m$ separately, but $\mathbf{A}_j^m$ directly, we have the additional storage requirement of $\mathbf{u}_1$.

Both, the data compression described in section 2.3 as well as the complexity reduction due to the hierarchical search have an impact on the perceptual quality. Experiments to verify the level of this impact, as well as the amount of obtainable data compression and complexity reduction are presented in the next section.

3. EXPERIMENTAL RESULTS

We evaluated the performance of the proposed data compression technique with experiments over the `CMU_ARCTIC` database from the Language Technologies Institute at Carnegie Mellon University[7]. The `CMU_ARCTIC` database was recorded specifically to be employed in corpus based speech synthesis. It includes datasets from two *US English* male speakers with identifiers `BDL` and `RMS`, two *US English* female speakers with identifiers `SLT` and `CLB`, one *Canadian English* male speaker with identifier `JMK`, one *Scottish English* male speaker with identifier `AWB` and one *Indian English* male speaker with identifier `KSP`. Each of the seven speaker subsets contains 1132 phonetically balanced English utterances. The utterances are roughly between one and four seconds long.

All speaker subsets of the corpus were employed in our study. The data was processed at a sampling rate of 16kHz. We divided the data into two strictly disjoint sets. 1117 utterances were used for the inventory design process and 15 utterances were used for the evaluation. Noise distortion was performed by adding *white* Gaussian noise at a signal-to-noise ratio (SNR) of 10 dB under consideration of the *active speech level* after ITU-T recommendation P.56.

3.1 Data Compression Results

Figure 2 illustrates how much data compression was achieved with the proposed method for a given *equivalent inventory reconstruction SNR* (EIN SNR_{def} after section 2.3). Results are listed separately for each speaker. The *remaining data size* refers to the relative storage requirement for the matrices $\tilde{\mathbf{U}}_j^m$ and $\tilde{\mathbf{\Phi}}_j^m$ from equation (17) in relation to the storage requirement for the uncompressed inventory. We already mentioned in section 2.3 that the decomposition of $\mathbf{A}_j^m$ into $\tilde{\mathbf{U}}_j^m$ and $\tilde{\mathbf{\Phi}}_j^m$ does not necessarily always result in data compression for all sub-clusters. For those sub-clusters for which the storage requirement for $\tilde{\mathbf{U}}_j^m$ and $\tilde{\mathbf{\Phi}}_j^m$ exceeded that of $\mathbf{A}_j^m$ we counted the storage requirement for the elements of $\mathbf{A}_j^m$ instead, plus the storage requirement for the dominant eigenvector $\mathbf{u}_1$ (which is required for the execution of the hierarchical search procedure after section 2.4).

A summary of average remaining data sizes over all speakers at various *equivalent inventory reconstruction SNR* levels are shown in table I. At an EIN SNR of 10 dB is is possible to reduce the inventory in average to around 12 % of its original size. Further compression is possible, however, at the expense of a more dramatic decline of perceptual quality of the system output (see section 3.3).

It should be emphasized that the reported compression levels are measured by simply counting the number of scalar values that need to be kept in memory. Significant further compression levels are very likely achievable with the introduction of suitable quantization schemes [7].

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

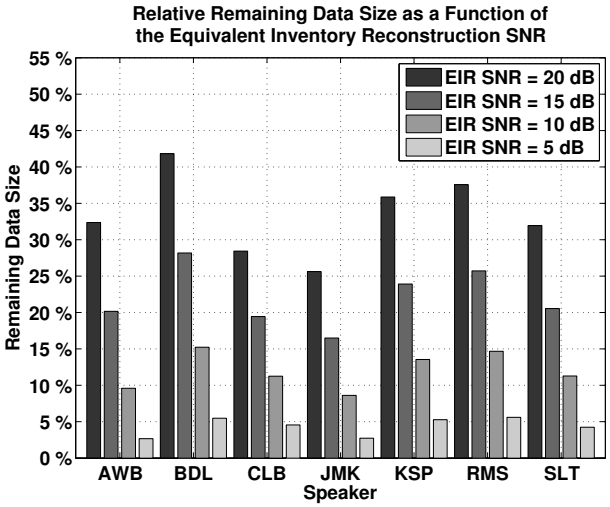

Figure 2. The remaining data sizes are shown as a percentage of the respective storage requirement for the full inventory. Date sizes are listed separately for each speaker. The average numbers across all speakers are shown in table I.

Table I

Average Remaining Data Size as a Function of the Equivalent Inventory Reconstruction SNR (EIR SNR).

EIR SNR	Average Remaining Data Size
20 dB	33.43 %
15 dB	22.13 %
10 dB	12.09 %
5 dB	4.40 %

3.2 Complexity Reduction

As pointed out in section 2.4 it is possible to also dramatically reduce the computational complexity of the proposed inventory based enhancement scheme with the employment of the hierarchical search instead of the exhaustive search for the best matching inventory unit from reference [5].

A precise evaluation of the computational complexity of a particular processing scheme is typically tied to its exact implementation details. Since there are many implementation schemes possible for the proposed procedure it was necessary (for the sake of a representative complexity analysis) to make the following assumption: All correlation procedures are performed with 512-tap *divide-and-conquer* FFTs (radix-2) [8]. For the execution of each FFT all multiplications and additions were counted, including the trivial ones. Correlations with signal lengths in excess of 512 taps were assumed to be processed with an *overlap-and-add* technique [8]. We assumed, furthermore, that each cluster m as well as each sub-cluster j within each cluster is equally probable to be chosen during the enhancement process.

The results of the complexity analysis, again as a function of the *equivalent inventory reconstruction SNR,* are shown in figure 3. The *remaining complexity* refers to the relative number of additions and multiplications required for the hierarchical search in relation to the corresponding operations

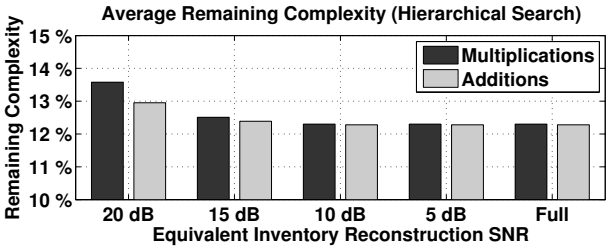

Figure 3. The hierarchical search proposed in section 2.4 provides a significant reduction in computational complexity. The (approximate) number of multiplications and additions that remains for the within-cluster search after section 2.4 is shown as a percentage of the respective number for the full inventory/full search system described in [5].

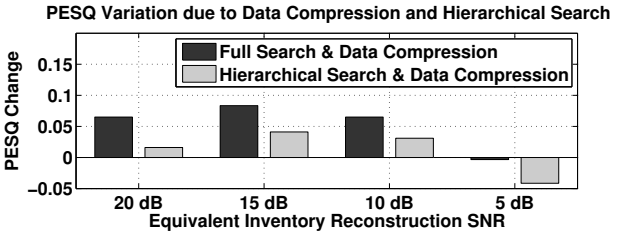

Figure 4. The performance of the proposed method in terms of PESQ scores is shown via the respective difference from the associated PESQ score of the full inventory/full search performance. PESQ scores were averaged over all testing utterances and all speakers.

required for the full search. It is clearly visible that the overall complexity can be reduced to about 12 % (i.e. a reduction of 88 % from the full search value) at an EIR SNR of 10 dB and below. The EIR SNR entry `FULL` refers to the case when no data compression is applied[8] and therefore the decoding of the inventory via equation (17) is not necessary. The slight complexity increase towards an EIR SNR of 20 dB and 15 dB is due to fact that at higher EIR SNR levels the additional complexity due to the evaluation of equation (17) is no longer negligible compared to the computation of the correlation values.

3.3 Perceptual Performance

Lastly, an objective quality assessment was performed with the *Perceptual Evaluation of Speech Quality* (PESQ) measure. The PESQ measure is an ITU recommendation developed by Rix *et. al.* [9] that is reported to correlate very well with *subjective quality* of speech[9]. We chose the PESQ measure to be able to report results that were consistent with the results reported in [5] for the full search approach with (almost) the same data set and with comparable processing conditions. Figure 4 shows the change in PESQ value for various compression and search conditions from the full inventory/full search procedure after [5] (baseline PESQ of 2.3). It is clearly visible that the proposed data compression as well as the hierarchical search do not lead to a significant reduction in PESQ scores, but rather to a slight increase.

[8] i.e. when the EIR SNR is ∞ dB.

[9] The PESQ measure was originally developed for the evaluation of speech coding algorithms. It has, nevertheless, been used to evaluate many speech enhancement methods as well.

This observation is also confirmed by informal listening tests with a few expert listeners who reported that the data compression between 20 dB EIR SNR and 10 dB EIR SNR led to a positive reduction in some of the high pitched musical noise that is present in the output of the full inventory/full search procedure. The reduction in musical noise is partially due a suppression of the spectral fine structure (especially at higher frequencies) of the re-synthesized speech. While this reduction in spectral fine structure was still rated as acceptable at an EIR SNR of 10 dB it was no longer rated acceptable at an EIR SNR of 5 dB.

4. CONCLUSIONS

We presented a method for a dramatic reduction in memory requirement and computational complexity for the inventory style speech enhancement scheme that was proposed by Xiao and Nickel in [5]. Experiments show that with an acceptable loss of spectral fine structure, yet an appreciable improvement in musical noise reduction, both the computational complexity as well as the memory requirements can be reduced to around 12 % of the corresponding requirements for the reference method [5]. Sound examples will be presented at the conference.

REFERENCES

[1] W. Jin, X. Liu, M. S. Scordilis, and L. Han, "Speech enhancement using harmonic emphasis and adaptive comb filtering," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 356–368, Feb. 2010.

[2] B. N. M. Laska, M. Bolić, and R. A. Goubran, "Particle filter enhancement of speech spectral amplitudes," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2155–2167, Nov. 2010.

[3] S. Jo and C. D. Yoo, "Psychoacoustically constrained and distortion minimized speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2099–2110, Nov. 2010.

[4] H. Ding, I. Y. Soon, and C. K. Yeo, "Over-attenuated components regeneration for speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 8, pp. 2004–2014, Nov. 2010.

[5] X. Xiao and R. M. Nickel, "Speech enhancement with inventory style speech resynthesis," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1243–1257, Aug. 2010.

[6] J. Ming, R. Srinivasan, and D. Crookes, "A corpus-based approach to speech enhancement from nonstationary noise," *Proceedings of Interspeech, Makuhari, Japan*, pp. 1097–1100, Sept. 2010.

[7] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communication Systems*, Wiley, 2004.

[8] R. E. Blahut, *Fast Algorithms For Digital Signal Processing*, Addison-Wesley, 1985.

[9] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ) - a new method for speech quality assessment of telephone networks and codecs," *Proceedings of ICASSP*, vol. 2, pp. 749–752, 2001.