PhD Thesis

Kernel PCA and Pre-Image Iterations for Speech Enhancement

Christina Leitner

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Austria

Advisors:
Assoc.Prof. Dipl.-Ing. Dr.mont. Franz Pernkopf
Univ.-Prof. Dipl.-Ing. Dr.techn. Gernot Kubin

Examiners:
Assoc.Prof. Dipl.-Ing. Dr.mont. Franz Pernkopf
Prof. Dr.-Ing. Rainer Martin

Graz, July 2013

EIDESSTATTLICHE ERKLÄRUNG

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am ....
(Unterschrift)

STATUTORY DECLARATION

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

....
date (signature)

Abstract

In this thesis, we present novel methods to enhance speech corrupted by noise. All methods are based on the processing of complex-valued spectral data. First, kernel principal component analysis (PCA) for speech enhancement is proposed. Subsequently, a simplification of kernel PCA, called pre-image iterations (PI), is derived. This method computes enhanced feature vectors iteratively by linear combination of noisy feature vectors. The weighting for the linear combination is found by a kernel function that measures the similarity between the feature vectors. The kernel variance is a key parameter for the degree of de-noising and has to be set according to the signal-to-noise ratio (SNR). Initially, PI were proposed for speech corrupted by additive white Gaussian noise. To be independent of knowledge about the SNR and to generalize to other stationary noise types, PI are extended by automatic determination of the kernel variance for white and colored noise. This enables a setting of the kernel variance without prior knowledge about the SNR. For colored noise this setting is frequency-dependent.

PI are executed on feature vectors extracted from the spectral representation. Analysis of PI shows that the convergence behavior of the feature vectors reveals information about the signal content. We use this information to segment the spectral representation in speech and non-speech regions and derive a mask for musical noise suppression in enhanced speech as a post-processing step.

We evaluate the proposed methods by listening, visual inspection of the spectrograms, by objective quality measures and the word accuracy of an automatic speech recognizer. Listening to the utterances and visual inspection of the spectrograms confirm the suppression of noise. No musical noise occurs, however, there is some residual noise around speech components. In terms of objective quality measures, the proposed methods achieve similar results as the generalized subspace method, spectral subtraction and the minimum mean-square error log-spectral amplitude estimator evaluated on speech corrupted by white noise. PI with automatic determination of the kernel variance achieve better results than the initial PI method. For colored noise, the performance of PI is better than the performance of the generalized subspace method, but weaker than the performance of the other two reference methods. In terms of automatic speech recognition, PI with automatic determination of the kernel variance achieve a performance superior to the generalized subspace method and similar to spectral subtraction, while the minimum mean-square error log-spectral amplitude estimator achieves higher recognition results.

Kurzfassung

In dieser Dissertation werden neue Methoden zur Qualitätsverbesserung von verrauschten Sprachaufnahmen vorgestellt. Alle Methoden basieren auf der Verarbeitung von komplexwertigen spektralen Daten. Als erste Methode wird die kernbasierte Hauptkomponentenanalyse für die Entrauschung der Sprachaufnahmen vorgeschlagen. Davon wird eine Vereinfachung abgeleitet, die wir Urbilditerationen bzw. *pre-image iterations* (PI) nennen. Diese Methode berechnet verbesserte Merkmalsvektoren auf iterative Weise aus linearen Kombinationen von verrauschten Merkmalsvektoren. Die Gewichtung der Linearkombinationen wird durch eine Kernfunktion festgelegt, die die Ähnlichkeit zwischen den Merkmalsvektoren bestimmt. Die Varianz der Kernfunktion ist ein wichtiger Parameter für den Grad der erreichten Rauschunterdrückung und muss abhängig vom Signal-Rauschabstand (SNR) gesetzt werden. Ursprünglich wurde die PI-Methode für additives weißes Gaußsches Rauschen vorgeschlagen. Um jedoch unabhängig von der Kenntnis des SNRs zu sein und um die Method für andere stationäre Geräuscharten zu verallgemeinern, wurde die PI-Methode mit einem Mechanismus zum automatischen Einstellen der Kernvarianz für weißes und farbiges Rauschen erweitert. Im Fall von farbigem Rauschen ist der Wert der Kernvarianz frequenzabhängig.

Die PI-Methode wird auf Merkmalsvektoren ausgeführt, die aus der spektralen Darstellung extrahiert sind. Die Analyse der PI zeigt, dass das Konvergenzverhalten Rückschlüsse auf den Inhalt des Signals zulässt. Wir verwenden diese Information um die spektrale Darstellung in Regionen mit und ohne Sprache zu segmentieren und um eine Maske zur Unterdrückung von *musical noise* für die Nachbearbeitung von entrauschten Sprachaufnahmen abzuleiten.

Wir evaluieren die vorgeschlagenen Methoden durch Anhören, visuelles Überprüfen der Spektrogramme, mit objektiven Qualitätsmaßen und der Worterkennungsrate eines automatischen Spracherkenners. Anhören der Sprachaufnahmen und das visuelle Überprüfen der Spektrogramme bestätigen die Unterdrückung des Rauschens. *Musical noise* tritt nicht auf, jedoch verbleiben Rauschkomponenten um die Sprachkomponenten herum. Für Sprachaufnahmen, die durch weißes Rauschen gestört sind, sind die erreichten Qualitätsmaße vergleichbar mit den Maßen der Referenzmethoden, der *generalized subspace*-Methode, der *spektralen Subtraktion* und dem *minimum mean-square error log-spectral amplitude estimator*. Die PI-Methode mit automatischem Setzen der Kernvarianz erreicht bessere Ergebnisse als die ursprünglich vorgeschlagene PI-Methode. Mit farbigem Rauschen sind the Qualitätsmaße von PI höher als die Maße der *generalized subspace*-Methode, jedoch niedriger als die Maße der zwei anderen Referenzmethoden. Die Spracherkennungsergebnisse für PI

mit automatischem Setzen der Varianz sind besser als die Ergebnisse der *generalized subspace*-Methode und ähnlich den Ergebnissen der spektralen Subtraktion. Der *minimum mean-square error log-spectral amplitude estimator* hingegen erreicht höhere Erkennungsraten.

Acknowledgment

First of all, I would like to thank Franz Pernkopf and Gernot Kubin for giving me the opportunity of pursuing the PhD studies at SPSC. Franz, for being my supervisor for this work, for contributing input and providing support, and for many proofread pages. Gernot, for encouraging my application, for interesting discussions and useful comments. I would like to thank Rainer Martin for valuable feedback and for his agreement to come to Graz.

In everyday life, I thank Bobi for being there with unbeatable optimism and for many small thinks that value so much, like preparing a heart-warming homecoming meal after exhausting travels. I wish you will lead me through many more tango nights. I am also grateful to our furry flatmate Schnurri for being hilarious and for showing what it means to be really relaxed – and that is quite contagious.

I would like to thank my family for the many things you do to support me. Just to name a few: Ursi for being an excellent traveling company and Agnes for the everyday matters a close-living family member has to deal with – and for being a demanding trainer. Mum and Dad for being there whenever help or support is needed.

Life would be boring without friends, so I would like to thank you, whether you are here or farther away. Thank you Benjamin, for never forgetting me in spite of the distance and much unanswered mail. Thank you Johanna and Gudrun, Gudrun and Karin, Birgit and Caro, Karin, Andi, Christoph, Christoph, and Christof for your company.

Spending a lot of time at our workplace, I would like to thank all my colleagues, especially Thomas, Michi, Sebastian, and Robert for sharing the office – with or without open window – and for numerous coffees and ice creams. Special thanks go as well to Anna for yoga and breakfast company and the encouragement to run.

Graz, July 2013 Christina Leitner

Contents

Abbreviations

APS	Artifact perceptual score
ASL	Active speech level
ASR	Automatic speech recognition
AWGN	Additive white Gaussian noise
BAS PD1	Bavarian Archive for Speech Signals Corpora PhonDat 1
CO	Combined pre-imaging
DAM	Diagnostic acceptability measure
DFT	Discrete Fourier transform
EVD	Eigenvalue decomposition
fwsegSNR	Frequency-weighted segmental SNR
IMCRA	Improved minima-controlled recursive averaging
IP	Iterative pre-imaging
IPS	Interference perceptual score
MIRS	Modified intermediate reference system
MMSE	Minimum mean-square error
MNS	Musical noise suppression
MOS	Mean opinion score
NF	Neighborhood filtering
NIP	Normalized iterative pre-imaging

NL	Non-local means
NLDF	Non-local diffusion filters
NP	Non-iterative pre-imaging
OLA	Overlap-add
OM-LSA	Optimally modified log-spectral amplitude
OPS	Overall perceptual score
PCA	Principal component analysis
PEASS	Perceptual evaluation methods for audio source separation
PESQ	Perceptual evaluation of speech quality
PI	pre-image iterations
PID	PI with determination of the kernel variance
PIDF	PI with frequency-dependent determination of the kernel variance
PSD	Power spectral density
RIP	Regularized iterative pre-imaging
RMSE	Root mean square error
SNR	Signal-to-noise ratio
STFT	Short-time Fourier transform
STSA	Short-time spectral amplitude
SVD	Singular value decomposition
TPS	Target perceptual score
VAD	Voice activity detection

Chapter 1

Introduction

In speech processing, speech is captured by a microphone in order to be stored or transmitted [1]. The presence of noise is often inevitable. For instance, it may not be possible to position a microphone close to the speaker or there is high-level surrounding sound, such as in traffic or in the presence of other speakers. Noise sources influence the quality and the intelligibility of transmitted speech. In telecommunications, reduced quality introduces listener fatigue and decreased intelligibility impedes the communication. For listeners with hearing aids the reduced intelligibility in the presence of noise constitutes an even bigger problem. Noise is not only problematic for humans, but also for man-to-machine communication. In telecommunications, the remote end may be equipped with an automatic speech recognition (ASR) system, or a mobile phone may provide ASR for voice dialing. In both scenarios, the recognition accuracy will most probably decrease if the signal is corrupted by noise [2].

In order to alleviate these problems, noise reduction or speech enhancement methods are applied. Ideally, noise should be attenuated and the speech components should be left unaffected. However, in practice, this is hardly possible. Noise reduction mostly comes along with speech distortion. Therefore, the objective of most enhancement algorithms is to reduce noise while keeping speech distortion as low as possible. Speech distortion mostly degrades the intelligibility. Unfortunately, the ideal case, where both quality and intelligibility are improved, is not reached by most speech enhancement algorithms.

Speech enhancement addresses many scenarios, depending on the noise type and the number of available microphones. The noise source can either be stationary, such as car noise, or non-stationary, such as babble noise. Noise can be correlated with the speech signal or not. It can be additive or the signal can be distorted by the reverberation in a room. Enhancement usually becomes easier when more than one microphone is used. For instance, if a microphone can be positioned close to the noise source, the signal can be used for adaptive noise canceling techniques [2]. Furthermore, microphone arrays allow for multichannel processing, which covers an own class of algorithms such as beamforming techniques [1].

In this work, we only consider uncorrelated stationary additive noise sources and signals captured by one microphone, such that only the noisy signal is available. As we only consider this noise condition, we will use the term speech enhancement equivalently to noise reduction. Our main objective is to improve the speech quality, which forms the basis for the used evaluation methods. Good intelligibility is desirable, however we do not explicitly focus on its optimization.

Most speech enhancement algorithms are applied after transformation of the time-domain signal. Usually, small segments of the time-domain signal, so-called frames, are transformed, for instance using the discrete Fourier transformation. The resulting transform coefficients – or spectral bins for the discrete Fourier transform (DFT) – are then modified according to the gain function of the given algorithm. In the case of the DFT, the gain function is usually applied on the magnitude of the complex-valued DFT coefficients. For inverse transformation to the time domain, the phase is required. Most speech enhancement algorithms do not estimate the phase of the clean speech signal but use the phase of the noisy speech signal. This can affect the speech quality at low signal-to-noise ratios [2].

The method proposed in this thesis is inspired by subspace methods which make use of principal component analysis (PCA) to attenuate noise components in the signal. The initial idea was to investigate if noise reduction can be improved by using non-linear techniques instead of (linear) PCA. In machine learning, the application of kernel methods constitutes a simple possibility to make linear algorithms non-linear. For this purpose, it must be possible to formulate the original algorithm in terms of inner products of feature vectors, which is the case for PCA. Then the inner product can be substituted by a non-linear kernel function and the algorithm becomes non-linear. The non-linear extension of PCA is known as *kernel PCA* [3, 4]. The kernel implicitly transforms the data to the so-called feature space where the data is processed. For de-noising, the processed data needs to be transformed back to the original input space. This is problematic if the transformation is non-linear, because there is possibly no one-to-one mapping between input and feature space. The data samples in input space that correspond to processed samples in feature space are called pre-images and the problem of finding the input space samples is therefore called the pre-image problem [5, 3].

In this work, we use an iterative pre-image method where the pre-image is computed by a linear combination of noisy feature vectors that are weighted by the kernel – which serves as similarity measure – and by weights derived from the projection step in kernel PCA. We show that the weights from the kernel PCA projection are negligible in comparison to the kernel weights and that they can be omitted for de-noising. Hence, de-noising is realized by the computation of linear combinations of noisy feature vectors based on their similarity, which is measured by the kernel. The proposed method is denoted *pre-image iterations* (PI) due to is derivation from an iterative pre-image method.

The feature vectors are quadratic patches extracted from the complex-valued spectro-temporal representation of speech utterances. Thus, we do not need the phase of the noisy speech signal for the inverse transformation to the time domain.

Furthermore, neighboring bins are processed jointly. This impedes the creation of musical noise, which mainly arises if the gains applied on neighboring bins vary largely. The feature extraction is similar to the feature extraction in image de-noising techniques, for instance kernel PCA for image de-noising [6] and the non-local means algorithm [7].

For evaluation, we use objective measures for speech quality, as the focus in this work lies on improvement of the quality (rather than the intelligibility). In addition, listening to the enhanced utterances and visual inspection of the spectrograms are used to derive conclusions about the effects of the applied methods. Finally, an automatic speech recognition experiment was conducted to test whether enhancement by pre-image iterations can improve the recognition ability of a pre-trained speech recognizer in comparison to noisy data.

In the following sections we will first give a short overview on existing speech enhancement methods. Then, we will discuss related work and highlight the scientific contributions of this work. Finally, we will give an outline of the thesis.

1.1 Overview on Speech Enhancement Algorithms

A vast number of algorithms for speech enhancement has been proposed in literature [2]. Among the first were spectral subtractive algorithms, which are probably the simplest. Spectral subtractive algorithms are based on the assumption that noise and speech are additive in the time domain. As a consequence, the speech and the noise spectrum are additive. To obtain the clean spectrum, the noise spectrum is estimated and subtracted from the noisy spectrum. The phase of the clean spectrum cannot be directly derived from the noisy spectrum, this is usually circumvented by subtracting magnitude values and using the phase of the noisy signal for synthesis. Boll describes magnitude spectral subtraction in [8].

Magnitude spectral subtraction can be extended to the power spectral domain. Based on the assumption that noise and speech are uncorrelated, the power spectrum of the noisy signal can be expressed as the sum of the power spectrum of the clean signal and the noise [2]. The inverse Fourier transform of the power spectrum is equal to the auto-correlation sequence. Therefore, after transforming the power spectra, spectral subtraction can also be executed in the correlation domain, as proposed by Weiss et al. in [9]. Furthermore, power spectral subtraction can be generalized by using other values for the exponent p instead of $p = 2$ for power spectral subtraction and $p = 1$ for magnitude spectral subtraction [2].

A major shortcoming of spectral subtraction is that it depends strongly on the accuracy of the noise estimate. If the noise estimate is poor, negative values occur in the clean spectrum estimate. This can be tackled by setting the corresponding frequency bins to zero. On the other hand, some bins may not be attenuated enough. If the frequency locations of these bins change from frame to frame, this is perceived as tonal artifacts of varying pitch. Due to the tonal quality this is referred to as *musical noise* [10]. Musical noise is not unique to spectral subtractive algorithms, it generally occurs if the time-varying gain function applied on the spectrum is poorly estimated by an enhancement algorithm. For spectral subtractive algorithms, Berouti et al. [10] proposed over-subtraction with spectral flooring to reduce musical noise. This algorithm creates less musical noise by filling the gaps between residual noise peaks.

While spectral subtraction aims to assess the clean signal by subtracting the noise estimate, the concept of statistical model-based algorithms is to find an estimate of the clean signal by formulation as an optimization problem. Usually, this is done in the spectral domain, i.e., either the complex or the magnitude DFT coefficients are determined. The optimization is based on the assumption that the DFT coefficients of speech and noise obey a certain probability density function that is incorporated in the optimization process. Mostly, a Gaussian probability distribution is used to model the real and imaginary parts of the clean DFT coefficients. The Gaussian distribution, however, does not optimally model speech DFT coefficients if a typical analysis window of 20-30 ms is used. Therefore the Gamma and Laplacian distribution have been proposed for more accurate modeling [2].

Among the statistical model-based algorithms, McAulay and Malpass [11] proposed maximum likelihood estimation to derive the clean DFT coefficients from the noisy DFT coefficients. Ephraim and Malah [12] proposed an estimator for the short-time spectral amplitude (STSA), i.e., the clean magnitude spectrum, based on the minimum mean-square error (MMSE). In addition to the magnitude, Ephraim and Malah also proposed estimators for the phase. These are, however, only optimal if the phase does not influence the magnitude estimate, as the magnitude estimate is already optimal. Besides the MMSE spectral amplitude estimator, other estimators were proposed that minimize the MMSE of the log-magnitude spectra or the p^{th} power spectrum (similar as for spectral subtraction). Alternatively to maximum likelihood and MMSE estimators, *maximum a posteriori* estimators were suggested [2, 13].

One further method based on the minimization of an error criterion is Wiener filtering [2, 14]. Generally, the goal is to find an optimal filter – the Wiener filter – that minimizes the estimation error between the filtered signal and a desired signal. Usually, the mean square of the estimation error is applied as optimization criterion. In speech enhancement, Wiener filters were first used by Lim and Oppenheim [15].

Subspace methods for speech enhancement represent another class of algorithms. They are based on linear algebra theory. The key idea is, that the clean speech signal only exists in a subspace of the noisy Euclidean space. If the Euclidean space can be decomposed into a subspace that contains noise and another subspace that contains the clean signal plus noise, an estimate of the clean signal can be retrieved by setting the components in the noise subspace to zero. The decomposition can be achieved by using orthogonal matrix-factorization techniques such as singular value decomposition (SVD) or eigenvalue decomposition (EVD), which is applied in PCA. Dendrinos et al. [16] proposed to use SVD on a matrix composed of time-domain amplitude values. The signal in the clean signal subspace is retrieved by first computing a *low-rank approximation* of the matrix [2]. Then, the de-noised time

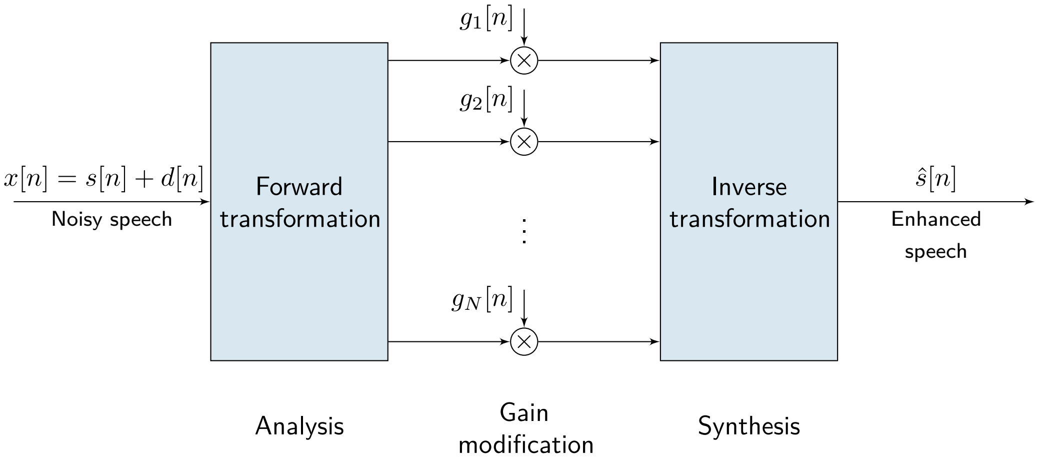


Figure 1.1: General structure of subspace methods and many other speech enhancement methods [2].

signal is synthesized by averaging the values in the approximation matrix. Ephraim and Van Trees [17] performed EVD on the signal covariance matrix. The noisy signal is projected to the clean signal subspace by applying a projection matrix composed of the principal eigenvectors on the vector containing the noisy signal. In practice, projecting the signal vector might not by sufficient for de-noising. Therefore, the signal vector is usually further modified, e.g., by some gain function $g_1[n], \ldots, g[n]_N$. Figure 1.1 shows the structure of most subspace methods. Generally, this structure also describes spectral subtractive algorithms. The main difference are the used transformations and the criteria to derive the gain functions. While in subspace methods the eigenvector matrix is used for transformation (this is also known as *Karhunen-Loève* transformation), spectral subtractive algorithms apply the Fourier transformation as forward transformation in the analysis block. In the synthesis block the respective inverse transformations are applied.

For many algorithms the availability of an accurate noise estimate is crucial. The simplest possibility is to estimate and update the noise when speech is absent using voice activity detection (VAD). This approach is mainly suitable for stationary or slowly changing noise types. In more realistic scenarios, such as in a restaurant, continuous updates of the noise estimate are preferable. Therefore, algorithms that continuously track the noise have been developed. These algorithms can be summarized in the following three classes: minimal-tracking algorithms, time-recursive averaging algorithms, and histogram-based algorithms [2].

Minimal tracking algorithms are based on the assumption that the power of the noisy speech signal in individual frequency bands often decreases to the power of the noise, even when speech is present. Consequently, a rough estimate of the noise can be found by tracking the minimum of the noisy signal power in each frequency band [2]. Two algorithms based on this assumption were proposed. The *minimum statistics* algorithm [18, 19] tracks the minimum within a finite analysis window, while the algorithm in [20] continuously tracks the minimum without application of a window. Both algorithms use the noisy signal power spectrum for the estimation. As this rapidly fluctuates, recursive smoothing is applied. The noise estimate of minimum statistics is biased towards values smaller than the mean noise power, therefore a bias correction factor was proposed for compensation.

Time-recursive algorithms estimate the noise spectrum as an average of past noise estimates and the current noisy signal spectrum. They make use of the observation that noise and speech power are not uniformly distributed over the frequency range. Therefore, each frequency bin of the noisy spectrum has a different effective signal-to-noise ratio (SNR). Whenever the SNR of a specific frequency bin is low, the estimate of the noise at this frequency bin is updated. Depending on the method, the weights for updating are based either on the SNR of each frequency bin or on the speech presence probability. One example of time-recursive averaging is the *improved minima-controlled recursive averaging* (IMCRA) algorithm [21]. This method incorporates speech presence probabilities in the spectral domain which we use for comparison of the proposed VAD derived from pre-image iterations.

Histogram-based techniques use the histograms of individual frequency bands of the noisy speech power spectrum for noise estimation. Usually, the histograms have either one or two modes, depending on the examined frequency band and other factors like signal duration and noisy type. In the two mode case, one mode corresponds to low energy segments, e.g., where speech is absent. The other mode corresponds to high energy segments such as voiced speech segments. Histogram-based techniques make use of the observation that the low energy mode is often the maximum value in the histogram. Therefore the maximum often indicates the noise level. A simple implementation of this algorithm is as follows: The noisy speech spectrum is smoothed to remove outliers. The histogram is created and the noise level corresponding to the maximum histogram bin is retrieved. This noise estimate is smoothed using first-order recursion [2].

Besides the noise estimate, many speech enhancement methods rely on SNR estimates. The *a priori* SNR denotes the true SNR between clean signal and noise, i.e., before noise is added to the clean signal. With *a posteriori* SNR we refer to the SNR between noisy signal and noise. A widely adopted approach to estimate the a priori SNR is the *decision-directed approach* [12], that derives the a priori SNR from a weighted average of the past a priori SNR estimate and the present a posteriori SNR estimate.

For a detailed description of speech enhancement methods, an extensive review is provided in [2].

1.2 Relation to Other Work

PCA provides the basis for this thesis. For speech de-noising, PCA has been applied in the context of subspace methods. Subspace methods have been proposed for white noise by Ephraim and Van Trees in [17], where they derive two estimators for the clean signal that minimize the speech distortion while keeping the level of the residual noise below a certain threshold (in time and frequency domain, respectively). Subspace methods are analyzed in the following theses: In [22], Hansen summarizes different estimators for the gain functions in a unified notation and analyzes their practical behavior. He discusses orthogonal rank-revealing matrix decomposition techniques and proposes a recursive decomposition algorithm that is based on updates after each new sample instead of frame-by-frame processing (as for instance applied in the algorithm proposed by Ephraim and Van Trees). This algorithm is numerically stable and achieves as good quality as the SVD-algorithm used for comparison. The subspace method by Ephraim and Van Trees has been generalized to colored noise by Hu in [23]. He proposes the *generalized subspace method* with built-in pre-whitening and derives linear estimators for the gain function in time and frequency domain. The suggested estimators lead to good performance on sentences of the TIMIT database corrupted by speech-shaped and multi-talker babble noise. In [24], Hermus derives an upper bound for the degree of de-noising in terms of SNR and proposes noise-shaping according to the MPEG-1 Layer 1 masking model, that aims to minimize speech distortion by only removing noise above the masking threshold of the speech signal. ASR is performed on utterances enhanced by different subspace methods and a performance gain in comparison to the noisy data is shown.

The kernel PCA and the pre-image iteration method proposed in this thesis are related to methods in both speech and image processing. Although subspace methods serve as idea for the proposed pre-image iteration method, there are substantial differences. Pre-image iterations are based on complex-valued features extracted from the sequence of short-time Fourier transforms, while subspace methods generally perform enhancement by transforming the time-domain signal and applying a gain function to the transform coefficients (as illustrated in Figure 1.1). Kernel PCA has been used in speech processing to extract robust features from reverberant speech in order to improve speech recognition rates [25]. This approach does not tackle the pre-image problem, as kernel PCA is used to extract the features which are directly fed into the speech recognizer, so no pre-image has to be computed.

In image processing, kernel PCA has been proposed to de-noise images [3, 6]. Mika et al. proposed an iterative solution to the pre-image problem when a Gaussian kernel is used and demonstrated its application to image de-noising [3]. The experiments are based on the USPS database [26], which contains small images of handwritten digits. Kernel PCA can also be applied to model images, i.e., for de-noising or compression. This, however, is only possible with a limited amount of data as the kernel grows quadratically with the number of training samples and manipulation and storage become problematic with a large amount of data. Kim et al. [6] proposed an iterative algorithm to apply kernel PCA on tasks involving a large number of training examples. They reported experiments on de-noising and super-resolution applications with a performance comparable to existing methods..

It can be shown that pre-image iterations are related to non-local neighborhood filtering, another de-noising technique used in image processing. Non-local neighborhood filtering is based on the assumption that it can be beneficial to process a signal value – i.e., the intensity of a pixel in the case of image de-noising – in a similar way as signal values with similar neighborhoods. While other de-noising algorithms often compute the value of the de-noised pixel solely based on the values of its surrounding pixels, non-local filters average over pixels that are located all over the image but have a similar neighborhood. This approach is favorable if images contain repetitive patterns such as textures [27, 7]. The non-local means algorithm is based on the same idea of exploiting neighborhoods [7]. With a specific choice of neighborhoods and processed sub-regions, pre-image iterations are equivalent to the first iteration of the non-local means algorithm except of the complex-valued feature vectors.

Although proven to be successful in image processing, non-local neighborhood filtering has only recently gained attention in speech processing. In [28], Talmon proposes the usage of non-local diffusion filters – which are related to non-local neighborhood filters – to suppress transient noise components in speech. However, in difference to pre-image iterations, the filter is not directly applied for de-noising. Non-local diffusion filters are used to gain an estimate of transient noise bursts, that is robust due to averaging over several instances of transients. This estimate is used as input for a noise suppression algorithm, jointly with an estimate for the background noise.

1.3 Scientific Contributions and Publications

This thesis covers the following contributions:

- Kernel PCA is applied for speech enhancement in the complex spectral domain. It is shown that additive white Gaussian noise is significantly reduced. The enhanced signals are not affected by musical noise, however, a buzz-like artifact occurs.

- We propose pre-image iterations derived from kernel PCA to de-noise speech corrupted by stationary white noise. This method is not affected by the buzz-like artifact in kernel PCA and free from musical noise. We extend the framework by automatic determination of the kernel variance, which serves as tuning parameter. This is abbreviated as PID – PI with determination of the kernel variance. Furthermore, we generalize pre-image iterations to the application on speech corrupted by stationary colored noise. We call this PIDF – PI with frequency-dependent determination of the kernel variance. The performance is evaluated in terms of objective quality measures. For white noise, the achieved scores are similar to the scores of the reference methods, namely the generalized subspace method, spectral subtraction and the MMSE log-STSA estimator. For colored noise, the scores achieved by PIDF are higher than the scores of the generalized subspace method but lower than the scores of the other two reference methods.

- A voice activity detector in the spectral domain is derived from pre-image iterations. The VAD can either be derived from noisy or enhanced signals. We show that the VAD can be used to derive a mask for musical noise suppression in enhanced speech. The application on enhanced signals allows for post-processing of speech enhanced by arbitrary enhancement methods.
- We perform ASR on the utterances enhanced by the proposed pre-image iteration methods. Compared to the noisy signals and to the signals enhanced by the generalized subspace method we obtain significantly better recognition rates with PID and PIDF in almost all SNR conditions.

The contributions in this thesis are divided between the author and co-workers as follows: The experimental work on speech enhancement for quality improvement was all conducted by the author. Several ideas arose from discussion with Franz Pernkopf and – to a minor amount – with Gernot Kubin. The work on ASR is joint work with Juan A. Morales Cordovilla, who contributed the trained speech models and the evaluation system, while the phonetic transcriptions and the grammar were provided by the author. The presented recognition results are from experiments conducted by the author. The following articles have been published during the course of this thesis:

- Christina Leitner, Franz Pernkopf, and Gernot Kubin, "Kernel PCA for speech enhancement," *12th Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 1221–1224, 2011.
- Christina Leitner and Franz Pernkopf, "The pre-image problem and kernel PCA for speech enhancement," in *Advances in Nonlinear Speech Processing*, vol. 7015 of *Lecture Notes in Computer Science*, pp. 199–206. 2011.
- Christina Leitner and Franz Pernkopf, "Speech enhancement using pre-image iterations," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4665–4668, 2012.
- Christina Leitner and Franz Pernkopf, "Musical noise suppression for speech enhancement using pre-image iterations," *International Conference on Systems, Signals and Image Procesing (IWSSIP)*, pp. 478–481, 2012.
- Christina Leitner and Franz Pernkopf, "Suppression of musical noise in enhanced speech using pre-image iterations," *20th European Signal Processing Conference (EUSIPCO)*, pp. 345–349, 2012.
- Christina Leitner and Franz Pernkopf, "Extension of pre-image speech denoising by voice activity detection using a bone conductive microphone," *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2012.
- Christina Leitner and Franz Pernkopf, "Generalization of pre-image iterations for speech enhancement," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7010–7014, 2013.

1.4 Outline of the Thesis

This thesis is organized as follows: In Chapter 2, we give a short introduction on kernel methods and the derivation of kernel PCA. The transformation of the processed samples in feature space back to the original input space is addressed by discussion of the pre-image problem. An overview of solutions to the pre-image problem is provided and several pre-image methods are compared experimentally for de-noising of synthetic data.

In Chapter 3, the application of kernel PCA for speech enhancement is presented. For de-noising, different pre-image methods are compared. Pre-image iterations derived from kernel PCA are proposed. Techniques to automatically determine the kernel variance are introduced and employed to generalize PI to colored noise. From the convergence behavior of pre-image iterations, information about the voice activity in the spectro-temporal representation can be derived. We show that this information can be used to perform musical noise suppression as post-processing of enhanced speech.

In Chapter 4, the *airbone* and the *Noizeus* databases used for the experiments are described. Evaluation is performed by state-of-the-art objective quality measures and by ASR. The applied measures – the perceptual evaluation of speech quality (PESQ) measure and the measures of the perceptual evaluation methods for audio source separation (PEASS) toolbox – are described and details about the speech recognizer are provided. Furthermore, the used reference methods are explained.

In Chapter 5, the results of kernel PCA, the variants of PI and the musical noise suppression methods in terms of objective quality measures and ASR are presented and discussed.

Chapter 6 concludes the thesis and provides a perspective on future work.

Chapter 2

Kernel Methods and the Pre-Image Problem

Kernel methods have gained considerable interest since the 1990ies [29]. One popular example are support vector machines. Kernels are defined as inner products in a so-called feature space. Kernel methods involve two processing steps: First, the data is mapped to a (possibly high-dimensional) feature space, then the algorithm at hand is executed in this feature space [4]. However, generally it is not necessary to compute the mapping of the data vectors. It is sufficient to evaluate the kernel of two data vectors in input space instead.

Given a set $\mathcal{X}$ of input samples $\mathbf{x}_i$, a kernel – or kernel function – is defined as follows

$$\begin{aligned} k : \mathcal{X} \times \mathcal{X} &\rightarrow \mathbb{R} \qquad (2.1) \\ (\mathbf{x}_i, \mathbf{x}_j) &\mapsto k(\mathbf{x}_i, \mathbf{x}_j), \end{aligned}$$

where the kernel $k(\cdot, \cdot)$ returns a scalar that describes the similarity of the samples $\mathbf{x}_i$ and $\mathbf{x}_j$. A simple example for such a similarity measure is an inner product. For example, the *canonical inner product* between two sample vectors is defined as

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \sum_{l=1}^{N} x_{il} x_{jl} = \mathbf{x}_i^T \mathbf{x}_j, \qquad (2.2)$$

where x_{il} is the l^{th} entry of $\mathbf{x}_i$ and N is the dimension of $\mathbf{x}_i$.

To compute the data vectors mapped to feature space $\mathcal{F}$, let us define the map

$$\begin{aligned} \mathbf{\Phi} : \mathcal{X} &\rightarrow \mathcal{F} \qquad (2.3) \\ \mathbf{x} &\mapsto \mathbf{\Phi}(\mathbf{x}). \end{aligned}$$

Now we are able to define the kernel as a similarity measure based on the inner product in $\mathcal{F}$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i)^T \mathbf{\Phi}(\mathbf{x}_j). \qquad (2.4)$$

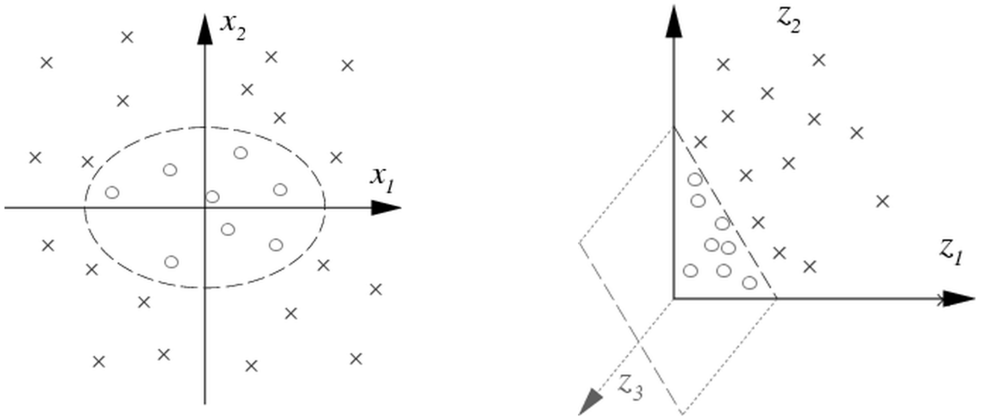


Figure 2.1: Binary classification problem with two classes (crosses and circles). In the two-dimensional input space the data is not linearly separable. After mapping to the three-dimensional feature space by the non-linear map $\mathbf{\Phi}(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ the data is separable by a hyperplane (figure from [4]).

By means of this relation, the inner product between the data vectors mapped to feature space can be evaluated by computing the kernel between the data vectors in input space. This is often referred to as the *kernel trick* [4]. For instance, assume the mapping $\mathbf{\Phi}(\mathbf{x}) = (x_1^2, x_2^2, \sqrt{2}x_1x_2)$ for the problem in Figure 2.1. The inner product in $\mathcal{F}$ is $\mathbf{\Phi}(\mathbf{x}_i)^T\mathbf{\Phi}(\mathbf{x}_j) = x_{i1}^2x_{j1}^2 + x_{i2}^2x_{j2}^2 + 2x_{i1}x_{i2}x_{j1}x_{j2}$. This can equivalently be expressed by the kernel $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j)^2$ in input space, i.e., $(\mathbf{x}_i^T\mathbf{x}_j)^2 = (x_{i1}x_{j1} + x_{i2}x_{j2})^2 = x_{i1}^2x_{j1}^2 + 2x_{i1}x_{j1}x_{i2}x_{j2} + x_{i2}^2x_{j2}^2 = \mathbf{\Phi}(\mathbf{x}_i)^T\mathbf{\Phi}(\mathbf{x}_j)$. As a consequence, any algorithm that can be formulated in terms of inner products can be generalized by substituting inner products with kernels. The freedom to choose the kernel and, equivalently, the map $\mathbf{\Phi}$ allows for adaption to a wide class of problems. Typically the map $\mathbf{\Phi}$ will be non-linear to generalize to non-linear problems. The simple example in Figure 2.1 illustrates how two datasets that are not linearly separable in input space become separable after a non-linear mapping to feature space. It has to be noted that in practice, the mapping will not be computed explicitly and often is even not known but only the kernel will be chosen according to the problem at hand. Usually the kernel is selected empirically.

2.1 Kernel Principal Component Analysis[1]

PCA is a widely used technique applied for dimensionality reduction, lossy data compression, feature extraction and data visualization [29, 31]. It is also referred to as the *Karhunen-Loève* transformation.

PCA is an orthogonal transformation of the coordinate system of the input data, i.e., the data is projected onto so-called *principal axes*. The new coordinates are

[1] This section is partly based on [30].

called *principal components*. Often the structure in data can be described with sufficient accuracy while using only a small number of principal components. This gives rise to applications like data compression where only components of large variance are retained as they are assumed to sufficiently cover the important information in the data. Similarly, for de-noising components with low variance are dropped as they are assumed to originate from noise [4, 32, 3].

PCA finds the principal axes by diagonalizing the estimated covariance matrix

$$\mathbf{S} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{x}_i \mathbf{x}_i^T, \tag{2.5}$$

of a set of M data samples $\mathbf{x}_i \in \mathbb{R}^N$, with $i = 1, \dots, M$, assuming zero mean $\sum_{i=1}^{M} \mathbf{x}_i = \mathbf{0}$ or, equivalently, centered samples. This leads to the eigenvalue equation

$$\lambda_l \mathbf{u}_l = \mathbf{S}\mathbf{u}_l, \tag{2.6}$$

which has to be solved for eigenvalues $\lambda_l \geq 0$ and nonzero eigenvectors $\mathbf{u}_l \in \mathbb{R}^N \setminus \{\mathbf{0}\}$, where $l = 1, \dots, N$ and the eigenvectors are normalized, i.e., $\mathbf{u}_l^T \mathbf{u}_l = 1$. The size of an eigenvalue λ_l corresponding to an eigenvector $\mathbf{u}_l$ is equivalent to the amount of variance in the direction of $\mathbf{u}_l$. Therefore, the principal components corresponding to the n largest eigenvalues cover the largest amount of variation in the data. Substituting (2.5) into (2.6) leads to

$$\lambda_l \mathbf{u}_l = \frac{1}{M}\sum_{i=1}^{M} (\mathbf{x}_i^T \mathbf{u}_l)\mathbf{x}_i. \tag{2.7}$$

This denotes a projection of the eigenvectors $\mathbf{u}_l$ with $\lambda_l \neq 0$ onto the samples $\mathbf{x}_i$. Consequently, all eigenvectors lie in the span of $\mathbf{x}_i, \dots, \mathbf{x}_M$, i.e., all $\mathbf{u}_l$ are linear combinations of $\mathbf{x}_i$ and can be written as expansions of $\mathbf{x}_i$ [29].

As PCA is linear, its ability to retrieve the structure within a given data set is limited. If the principal components of variables are non-linearly related to the input variables, a non-linear feature extractor is more suitable. This is realized by kernel PCA [4, 3].

To derive kernel PCA from standard PCA, let us assume a mapping $\mathbf{\Phi}(\mathbf{x})$ from input space $\mathcal{X}$ to feature space $\mathcal{F}$, as given in (2.3). As before, we assume that the data is centered in feature space $\sum_{i=1}^{M} \mathbf{\Phi}(\mathbf{x}_i) = \mathbf{0}$. In feature space, the estimated covariance matrix is

$$\mathbf{C} = \frac{1}{M}\sum_{i=1}^{M} \mathbf{\Phi}(\mathbf{x}_i)\mathbf{\Phi}(\mathbf{x}_i)^T. \tag{2.8}$$

To diagonalize the covariance matrix we have to solve the eigenvalue equation

$$\lambda_k \mathbf{v}_k = \mathbf{C}\mathbf{v}_k \tag{2.9}$$

for eigenvalues $\lambda_k \geq 0$ and non-zero eigenvectors $\mathbf{v}_k \in \mathcal{F} \setminus \{\mathbf{0}\}$, $\mathbf{v}_k^T \mathbf{v}_k = 1$. Equivalently to (2.7), all eigenvectors $\mathbf{v}_k$ that solve this equation lie in the span of

$\mathbf{\Phi}(\mathbf{x}_1), \ldots, \mathbf{\Phi}(\mathbf{x}_M)$. Therefore, each eigenvector $\mathbf{v}_k$ can be written as linear combination of the mappings $\mathbf{\Phi}(\mathbf{x}_i)$ using the coefficients $\alpha_{k1}, \ldots, \alpha_{kM}$

$$\mathbf{v}_k = \sum_{i=1}^{M} \alpha_{ki} \mathbf{\Phi}(\mathbf{x}_i). \tag{2.10}$$

Substituting(2.8) and (2.10) into (2.9) leads to

$$\lambda_k \sum_{i=1}^{M} \alpha_{ki} \mathbf{\Phi}(\mathbf{x}_i) = \frac{1}{M} \sum_{j=1}^{M} \mathbf{\Phi}(\mathbf{x}_j) \mathbf{\Phi}(\mathbf{x}_j)^T \sum_{i=1}^{M} \alpha_{ki} \mathbf{\Phi}(\mathbf{x}_i) \tag{2.11}$$

for all $k = 1, \ldots, M$. To enable an expression in terms of kernel functions we multiply both sides by $\mathbf{\Phi}(\mathbf{x}_k)^T$ such that

$$\lambda_k \sum_{i=1}^{M} \alpha_{ki} \mathbf{\Phi}(\mathbf{x}_k)^T \mathbf{\Phi}(\mathbf{x}_i) = \frac{1}{M} \sum_{j=1}^{M} \mathbf{\Phi}(\mathbf{x}_k)^T \mathbf{\Phi}(\mathbf{x}_j) \sum_{i=1}^{M} \alpha_{ki} \mathbf{\Phi}(\mathbf{x}_j)^T \mathbf{\Phi}(\mathbf{x}_i) \tag{2.12}$$

for all $k = 1, \ldots, M$. The multiplication of the mappings $\mathbf{\Phi}(\mathbf{x}_i)^T \mathbf{\Phi}(\mathbf{x}_j)$ can be expressed as kernel in terms of input samples $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{\Phi}(\mathbf{x}_i)^T \cdot \mathbf{\Phi}(\mathbf{x}_j)$. Now, let us define an $M \times M$ matrix $\mathbf{K}$ called *kernel matrix* or *Gram matrix* with the entries

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j). \tag{2.13}$$

Then, Equation (2.12) can be reformulated as

$$M \lambda_k \mathbf{K} \boldsymbol{\alpha}_k = \mathbf{K}^2 \boldsymbol{\alpha}_k, \tag{2.14}$$

where $\boldsymbol{\alpha}_k$ is the k^{th} eigenvector with the entries $\alpha_{k1}, \ldots, \alpha_{kM}$. The eigenvectors of this system equivalently solve the eigenvalue problem

$$M \lambda_k \boldsymbol{\alpha}_k = \mathbf{K} \boldsymbol{\alpha}_k. \tag{2.15}$$

To find the eigenvectors $\boldsymbol{\alpha}_k$ the matrix $\mathbf{K}$ has to be diagonalized. Let us denote the eigenvalues of $\mathbf{K}$ in the following by $\lambda_1, \ldots, \lambda_M$ (which are equivalent to the eigenvalues $M\lambda_k$ solving (2.15)). By requiring a normalization of the eigenvectors in feature space, i.e., $\mathbf{v}_k^T \mathbf{v}_k = 1$, we can derive the normalization condition for the eigenvectors $\boldsymbol{\alpha}_k$

$$\begin{aligned} 1 &= \mathbf{v}_k^T \mathbf{v}_k = \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_{ki} \alpha_{kj} \mathbf{\Phi}(\mathbf{x}_i)^T \mathbf{\Phi}(\mathbf{x}_j) \\ &= \sum_{i=1}^{M} \sum_{j=1}^{M} \alpha_{ki} \alpha_{kj} K_{ij} = \boldsymbol{\alpha}_k^T \mathbf{K} \boldsymbol{\alpha}_k = \lambda_k \boldsymbol{\alpha}_k^T \boldsymbol{\alpha}_k, \end{aligned} \tag{2.16}$$

using (2.10) and (2.15).

The projection of a test sample $\mathbf{x}$ onto the eigenvectors $\mathbf{v}_k$ in $\mathcal{F}$ can then be determined as

$$\beta_k = (\mathbf{v}_k)^T \mathbf{\Phi}(\mathbf{x}) = \sum_{i=1}^{M} \alpha_{ki} \mathbf{\Phi}(\mathbf{x}_i)^T \mathbf{\Phi}(\mathbf{x}) = \sum_{i=1}^{M} \alpha_{ki} k(\mathbf{x}_i, \mathbf{x}). \tag{2.17}$$

In summary, to project $\mathbf{x}$ onto the eigenvectors $\mathbf{v}_k$ in $\mathcal{F}$ the following steps are required: (i) compute the kernel matrix $\mathbf{K}$, (ii) compute its eigenvectors $\boldsymbol{\alpha}_k$ and normalize them using (2.16), (iii) project the data sample $\mathbf{x}$ using (2.17). The computation only requires evaluation of kernels, the evaluation of the map $\mathbf{\Phi}(\mathbf{x})$ is not necessary.

2.1.1 Centering

Until so far, we have assumed that the data in feature space is centered. This can easily be ensured in input space $\mathcal{X}$, but is harder to achieve in feature space $\mathcal{F}$, as we usually do not explicitly compute the mapped data and therefore the quantity $\sum_{i=1}^{M} \mathbf{\Phi}(\mathbf{x}_i)$ cannot be assessed. However, as shown in [4, 32] centering can be done by modifying the kernel matrix $\mathbf{K}$ such that the *centered kernel matrix* $\tilde{\mathbf{K}}$ is

$$\tilde{\mathbf{K}} = \mathbf{K} - \mathbf{1}_M \mathbf{K} - \mathbf{K} \mathbf{1}_M + \mathbf{1}_M \mathbf{K} \mathbf{1}_M, \tag{2.18}$$

where $\mathbf{1}_M$ is an $M \times M$ matrix with all entries equal to $1/M$. The eigenvectors $\boldsymbol{\alpha}_k$ can then be computed by diagonalizing $\tilde{\mathbf{K}}$ instead of $\mathbf{K}$.

2.1.2 Kernel PCA for De-noising

To de-noise data, we assume that the directions of eigenvectors corresponding to small eigenvalues only contain information about noise, as small eigenvalues denote small variances. In contrast, eigenvectors corresponding to large eigenvalues are assumed to contain relevant information, e.g., speech. Therefore, the data sample $\mathbf{\Phi}(\mathbf{x})$ is projected onto the eigenvectors $\mathbf{v}_k$ corresponding to the n largest eigenvalues while the directions of small eigenvalues are dropped to remove the noise [3]. To reconstruct the mapping $\mathbf{\Phi}(\mathbf{x})$ after projection we define a projection operator P_n that is given as

$$P_n \mathbf{\Phi}(\mathbf{x}) = \sum_{k=1}^{n} \beta_k \mathbf{v}_k, \tag{2.19}$$

where the eigenvectors are assumed to be ordered by decreasing eigenvalue size. Consequently, $P_n \mathbf{\Phi}(\mathbf{x})$ is a linear combination of the first n eigenvectors $\mathbf{v}_k$ using the projections β_k of (2.17) as weights. In case of using all $\mathbf{v}_k$, the data sample after projection equals the original data sample $P_n \mathbf{\Phi}(\mathbf{x}) = \mathbf{\Phi}(\mathbf{x})$.

The drawback of de-noising in feature space is that in common applications the de-noised data is required in input space. The problem of finding the samples in input space that map to the projected samples in feature space is called the *pre-image problem.* This is addressed in the next section.

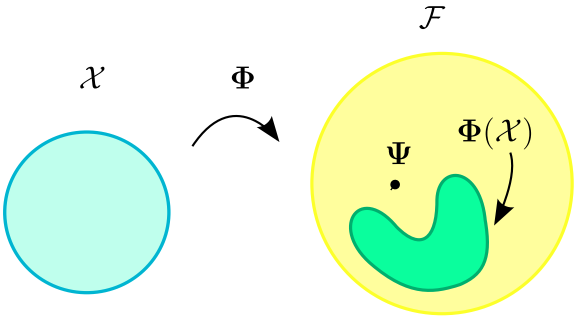


Figure 2.2: The pre-image problem: Points like $\mathbf{\Psi}$, which lie in the span of mapped input samples, are not necessarily images of input samples, i.e., they do not belong to the subspace $\mathbf{\Phi}(\mathcal{X})$ (figure after [4]).

2.2 Kernel PCA and the Pre-image Problem

With kernel methods, generally all computations in feature space $\mathcal{F}$ are done implicitly. Consequently, the solutions of kernel algorithms are expressed as expansions $\mathbf{\Psi}$ in terms of mapped input data – such as the eigenvectors $\mathbf{v}_k$ for kernel PCA (c.f. (2.10)),

$$\mathbf{\Psi} = \sum_{i=1}^{M} \alpha_i \mathbf{\Phi}(\mathbf{x}_i). \tag{2.20}$$

The map $\mathbf{\Phi}$ is usually non-linear and therefore not necessarily invertible. Hence, it cannot generally be assured that each expansion in feature space $\mathbf{\Psi}$ has a *pre-image* under $\mathbf{\Phi}$, i.e., a sample $\mathbf{z} \in \mathcal{X}$ such that $\mathbf{\Phi}(\mathbf{z}) = \mathbf{\Psi}$. This is illustrated in Figure 2.2). The pre-image problem has widely been studied and several solutions have been proposed [3, 33, 34, 35, 36, 37].

If the pre-image exists and if the kernel $k(\cdot,\cdot)$ is an invertible function f_k then the pre-image can be computed by

$$\mathbf{z} = \sum_{i=1}^{N} f_k^{-1}\left(\sum_{j=1}^{M} \alpha_j k(\mathbf{x}_j, \mathbf{e}_i)\right)\mathbf{e}_i \tag{2.21}$$

where $\mathbf{e}_1, \ldots, \mathbf{e}_N$ is any orthonormal basis of the input space [4, 32]. This is based on the fact that $\mathbf{\Phi}(\mathbf{z})$ is an expansion of the mappings in $\mathcal{F}$ as given in (2.20) and that $\mathbf{z}$ can be written as expansion of the orthonormal basis vectors, i.e., $\mathbf{z} = \sum_{i=1}^{N}(\mathbf{z}^T\mathbf{e}_i)\mathbf{e}_i$. Equation (2.21) returns the exact pre-image, because $\mathbf{z}$ is given by

$$\begin{aligned}
\mathbf{z} &= \sum_{i=1}^{N} f_k^{-1}\left(\sum_{j=1}^{M} \alpha_j k(\mathbf{x}_j, \mathbf{e}_i)\right)\mathbf{e}_i = \sum_{i=1}^{N} f_k^{-1}\left(\sum_{j=1}^{M} \alpha_j \mathbf{\Phi}(\mathbf{x}_j)^T \mathbf{\Phi}(\mathbf{e}_i)\right)\mathbf{e}_i \qquad (2.22)\\
&= \sum_{i=1}^{N} f_k^{-1}\left(\mathbf{\Phi}(\mathbf{z})^T\mathbf{\Phi}(\mathbf{e}_i)\right)\mathbf{e}_i = \sum_{i=1}^{N} f_k^{-1}\left(k(\mathbf{z}, \mathbf{e}_i)\right)\mathbf{e}_i = \sum_{i=1}^{N}(\mathbf{z}^T\mathbf{e}_i)\mathbf{e}_i.
\end{aligned}$$

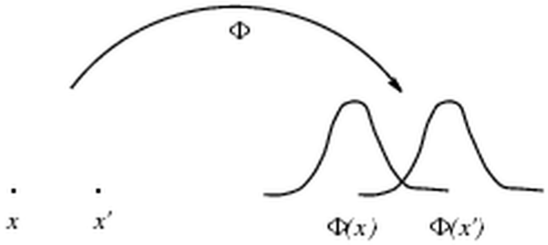


Figure 2.3: Visualization of the feature map $\mathbf{\Phi} : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}, \mathbf{x} \mapsto k(\cdot, x)$ using a Gaussian kernel. Each sample in feature space $\mathbf{\Phi}(\mathbf{x})$ is represented by a *function* sitting on the sample $\mathbf{x}$ (figure from [4]).

One example for the computation of the pre-image based on an invertible function are polynomial kernels

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T \mathbf{x}_j + c)^d, \text{ where } c \geq 0 \text{ and } d \text{ odd.} \tag{2.23}$$

However, in many cases no pre-image *exists*. For example, let us define a map $\mathbf{\Phi}$ from $\mathcal{X}$ into the space of functions $\mathbb{R}^{\mathcal{X}} := \{f : \mathcal{X} \to \mathbb{R}\}$, i.e., $\mathbf{\Phi} : \mathcal{X} \to \mathbb{R}^{\mathcal{X}}, \mathbf{x} \mapsto k(\cdot, \mathbf{x})$. This means that each sample is turned into a function on the domain $\mathcal{X}$. Under this map, pre-images only exist for functions in feature space that can be written as $k(\cdot, \mathbf{x})$. To give a specific example, consider the Gaussian kernel

$$k(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{c}\right), \tag{2.24}$$

where c is the kernel variance. Using the map $\mathbf{\Phi}$, each input is mapped to a Gaussian centered on this point (see Figure 2.3). An arbitrary expansion of Gaussians has no pre-image (except of trivial cases with one term), because a Gaussian cannot be written as linear combination of Gaussians centered at different points, but only a Gaussian would have an exact pre-image with this map [4].

The next section summarizes methods proposed in the literature that aim to solve the pre-image problem for Gaussian kernels and kernel PCA with regard to de-noising. Subsequently, an experimental evaluation on a subset of methods using synthetic data is given.

2.2.1 Overview on Pre-Image Methods

In the case of applying kernel PCA with a Gaussian kernel, one solution for the pre-image problem is to approximate the pre-image $\mathbf{z}$ by minimizing the Euclidean distance between $\mathbf{\Phi}(\mathbf{z})$ and the projection in feature space $P_n\mathbf{\Phi}(\mathbf{x})$

$$\rho(\mathbf{z}) = \|\mathbf{\Phi}(\mathbf{z}) - P_n\mathbf{\Phi}(\mathbf{x})\|^2. \tag{2.25}$$

Mika et al. [3] showed that for kernels that satisfy $k(\mathbf{x}, \mathbf{x}) = 1$ for all $\mathbf{x} \in \mathcal{X}$ (such as the Gaussian kernel) the minimization of $\rho(\mathbf{z})$ can be done by fixed point iterations.

To minimize $\rho(\mathbf{z})$, we reformulate (2.25) to

$$\begin{aligned}\rho(\mathbf{z}) &= \mathbf{\Phi}(\mathbf{z})^T\mathbf{\Phi}(\mathbf{z}) - 2\mathbf{\Phi}(\mathbf{z})^T P_n\mathbf{\Phi}(\mathbf{x}) + (P_n\mathbf{\Phi}(\mathbf{x}))^T P_n\mathbf{\Phi}(\mathbf{x}) && (2.26)\\ &= \mathbf{\Phi}(\mathbf{z})^T\mathbf{\Phi}(\mathbf{z}) - 2\mathbf{\Phi}(\mathbf{z})^T \sum_{k=1}^{n} \beta_k \mathbf{v}_k + \Omega, && (2.27)\end{aligned}$$

by using (2.19) and the definition of β_k in (2.17). The terms independent of $\mathbf{z}$ are replaced by Ω. Expanding the eigenvector $\mathbf{v}_k$ and substituting the inner products of mappings by kernels leads to

$$\begin{aligned}\rho(\mathbf{z}) &= \mathbf{\Phi}(\mathbf{z})^T\mathbf{\Phi}(\mathbf{z}) - 2\mathbf{\Phi}(\mathbf{z})^T \sum_{k=1}^{n} \beta_k \sum_{i=1}^{M} \alpha_{ki}\mathbf{\Phi}(\mathbf{x}_i) + \Omega && (2.28)\\ &= k(\mathbf{z},\mathbf{z}) - 2\sum_{k=1}^{n} \beta_k \sum_{i=1}^{M} \alpha_{ki} k(\mathbf{z},\mathbf{x}_i) + \Omega. && (2.29)\end{aligned}$$

As the term $k(\mathbf{z},\mathbf{z})$ is constant and Ω independent of $\mathbf{z}$, we need to maximize

$$J = \sum_{k=1}^{n} \beta_k \sum_{i=1}^{M} \alpha_{ki} k(\mathbf{z},\mathbf{x}_i) = \sum_{i=1}^{M} \gamma_i k(\mathbf{z},\mathbf{x}_i), \tag{2.30}$$

where $\gamma_i = \sum_{k=1}^{n} \beta_k \alpha_{ki}$ and $\sum_{i=1}^{M} \gamma_i \mathbf{\Phi}(\mathbf{x}_i) = \sum_{i=1}^{n} \beta_k \mathbf{v}_k$ equals the projection $P_n\mathbf{\Phi}(\mathbf{x})$ in feature space. For kernels of the form $k(\|\mathbf{z}-\mathbf{x}_i\|^2)$, e.g., Gaussian kernels, the gradient with respect to $\mathbf{z}$ evaluates to

$$\nabla_{\mathbf{z}} J = 2 \cdot \sum_{i=1}^{M} \gamma_i k'(\|\mathbf{z}-\mathbf{x}_i\|^2)(\mathbf{z}-\mathbf{x}_i), \tag{2.31}$$

where $k'(\cdot,\cdot)$ denotes the derivative of $k(\cdot,\cdot)$. Setting the above equation to zero leads to the extremum

$$\mathbf{z} = \frac{\sum_{i=1}^{M} \gamma_i k'(\|\mathbf{z}-\mathbf{x}_i\|^2)\mathbf{x}_i}{\sum_{i=1}^{M} \gamma_i k'(\|\mathbf{z}-\mathbf{x}_i\|^2)}. \tag{2.32}$$

For the Gaussian kernel this results in

$$\mathbf{z} = \frac{\sum_{i=1}^{M} \gamma_i \exp(-\|\mathbf{z}-\mathbf{x}_i\|^2/c)\mathbf{x}_i}{\sum_{i=1}^{M} \gamma_i \exp(-\|\mathbf{z}-\mathbf{x}_i\|^2/c)}. \tag{2.33}$$

The Gaussian kernel is smooth and therefore we assume that there is a neighborhood around the extremum of (2.30) where the denominator of (2.33) is $\neq 0$. Hence, we can execute (2.33) iteratively such that

$$\mathbf{z}^{t+1} = \frac{\sum_{i=1}^{M} \gamma_i k(\mathbf{z}^t,\mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{M} \gamma_i k(\mathbf{z}^t,\mathbf{x}_i)}, \tag{2.34}$$

where t denotes the iteration index. The weighting coefficients γ_i contribute information about the projection and the kernel provides a weight corresponding to the similarity between the pre-image $\mathbf{z}$ and the data samples $\mathbf{x}_i$. Note that the resulting pre-image $\mathbf{z}$ is always a linear combination of the input data $\mathbf{x}_i$. This algorithm is sensitive to initialization which, however, can be tackled by reinitializing with different values.

Several variations of this iterative pre-image solution were proposed. Kwok and Tsang [33] suggested to use normalized weighting coefficients in (2.34) to account for centering

$$\tilde{\gamma}_i = \gamma_i + 1/M(1 - \sum_{m=1}^{M} \gamma_m). \tag{2.35}$$

Abrahamsen and Hansen [34] further extended the method by a regularization term

$$\mathbf{z}_j^{t+1} = \frac{\frac{2}{c}\sum_{i=1}^{M} \tilde{\gamma}_i k(\mathbf{z}_j^t, \mathbf{x}_i)\mathbf{x}_i + \eta \mathbf{x}_j}{\frac{2}{c}\sum_{i=1}^{M} \tilde{\gamma}_i k(\mathbf{z}_j^t, \mathbf{x}_i) + \eta}, \tag{2.36}$$

where η is a non-negative regularization parameter and $\mathbf{x}_j$ is the noisy sample. They show that the method is more stable than the method of Mika et al.

In our experiments, we compare these three methods. In addition, we test the non-iterative pre-image method by Honeine and Richard [35], that preserves inner product measures in both spaces using least square techniques. A pre-image is computed by

$$\mathbf{z}_j = \mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X} - \eta\mathbf{K}^{-1})\tilde{\boldsymbol{\gamma}}, \tag{2.37}$$

where $\mathbf{X} = [\mathbf{x}_1\, \mathbf{x}_2\, \ldots\, \mathbf{x}_M]$, η is a regularization parameter and $\tilde{\boldsymbol{\gamma}} = [\tilde{\gamma}_1\, \tilde{\gamma}_2\, \ldots\, \tilde{\gamma}_M]^T$ with $\tilde{\gamma}_i$ from (2.35). This method does not require the computation of distances but relies only on inner products in input space and on kernel values in feature space. Furthermore it is numerically stable.

Further suggested methods comprise a non-iterative method of Kwok and Tsang [33] that finds the pre-image based on distance constraints in the feature space. Rathi et al. [36, 38] relaxed the method of Mika et al. by an approximation that allows for direct computation without iterations. Zheng et al. [37] proposed a two-step method, where the pre-image is modeled by a weighted combination of the data samples and the weights are learned by an optimization function that incorporates convexity constraints and a penalty function.

2.2.2 Experimental Comparison of Pre-Image Methods[2]

To compare the behavior of different pre-image methods, four methods were tested on synthetic data sets for de-noising with and without centering similar as in [35]. The following data sets were used:

- The *square dataset* consists of samples on a 1 × 1 square, where the samples for each edge are drawn from a uniform distribution and corrupted by additive white Gaussian noise of variance 0.01.

[2] This section is based on [39].

				Centered			
Dataset	n	c	Noisy	IP	NIP	RIP	NP
Square	6	0.25	0.1417 ± 0.0005	0.1242 ± 0.0005	0.1173 ± 0.0004	0.1168 ± 0.0004	0.1168 ± 0.0004
Sine	15	0.10	0.0711 ± 0.0005	0.0629 ± 0.0005	0.0600 ± 0.0004	0.0602 ± 0.0004	0.0602 ± 0.0004
Spiral	20	0.10	0.1004 ± 0.0005	0.0907 ± 0.0005	0.0887 ± 0.0004	0.0875 ± 0.0004	0.0875 ± 0.0004
Complex	6	0.25	0.1420 ± 0.0005	0.1233 ± 0.0005	0.1164 ± 0.0004	0.1161 ± 0.0004	0.1161 ± 0.0004

				Uncentered			
Dataset	n	c	Noisy	IP	NIP	RIP	NP
Square	6	0.25	0.1417 ± 0.0005	0.1227 ± 0.0004	0.1233 ± 0.0004	0.1198 ± 0.0004	0.1198 ± 0.0004
Sine	15	0.10	0.0711 ± 0.0005	0.0603 ± 0.0004	0.0603 ± 0.0004	0.0603 ± 0.0004	0.0603 ± 0.0004
Spiral	20	0.10	0.1004 ± 0.0005	0.0899 ± 0.0004	0.0900 ± 0.0004	0.0880 ± 0.0004	0.0880 ± 0.0004
Complex	6	0.25	0.1420 ± 0.0005	0.1225 ± 0.0004	0.1230 ± 0.0004	0.1196 ± 0.0004	0.1196 ± 0.0004

Table 2.1: RMSE and standard deviation for different pre-image methods on synthetic data. n denotes the number of components used for projection and c is the variance of the Gaussian kernel.

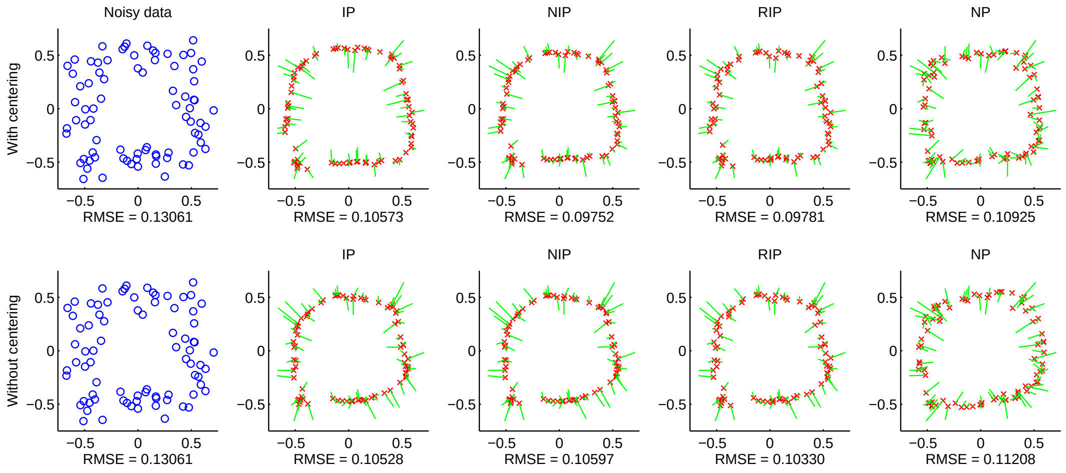


Figure 2.4: De-noising with different pre-image methods on centered (top) and uncentered (bottom) data of the square dataset with the kernel variance $c = 0.25$ and $n = 6$ components for projection. The green lines illustrate the distance between de-noised (red) and noisy samples.

- The *sine dataset* is specified by samples with the coordinates $(x, \sin(2\pi x))$ with x uniformly distributed on the interval $[0, 4\pi]$ plus additive white Gaussian noise of variance 0.0025.
- The *spiral dataset* is given by samples with the coordinates $(At\cos(t), At\sin(t))$ where $A = 0.1$ and t is uniformly distributed on the interval $[0, 4\pi]$. White Gaussian noise of variance 0.005 is added.
- The *complex-valued square dataset* is created in a similar way as the square dataset, however here the real part of the complex number corresponds to the first coordinate and the imaginary part to the second.

The following pre-image methods are compared on all datasets:

1. Iterative pre-imaging (IP) in (2.34).
2. Normalized iterative pre-imaging (NIP) : IP with normalization of the weighting coefficients as given in (2.35).
3. Regularized iterative pre-imaging (RIP) in (2.36).
4. Non-iterative pre-imaging (NP) (2.37).

For evaluation the root mean squared error (RMSE) between reconstructed samples and noise-free reference samples is computed. For each dataset the RMSE is averaged over 100 realizations. Table 2.1 shows selected results for the four datasets with and without centering using the IP, NIP, RIP, and the NP method. Figure 2.4 illustrates the de-noising and projection onto 6 principal components for one realization of the square dataset. Figure 2.5 shows de-noising for the sine and the spiral dataset with projection on 15 and 20 components, respectively (plots for uncentered data are omitted due to their similarity).

From the experiments, it can be concluded that NIP and RIP yield the best results. In contrast to IP, these methods perform normalization of the weighting coefficients which seems to be necessary to achieve good reconstruction quality of the pre-image. NP does not perform as good. It has to be noted that our experiment is different from [35], because we use the same data for training and testing – i.e., for projection and pre-image reconstruction – while they use different datasets. We encountered no problems of stability of the iterative algorithms, as we always use the noisy sample for initialization which seems to be very robust. Since the pre-image methods are used on complex-valued data in the case of speech enhancement we performed one further experiment on the *complex-valued square dataset.* The results are shown in Table 2.1 and illustrated in Figure 2.6. The example demonstrates that the pre-image methods can be applied to complex-valued data as well.

In literature, centering is noted to be important for kPCA. In our experimental setup with synthetic data we could, however, not observe any significant difference in the performance with and without centering.

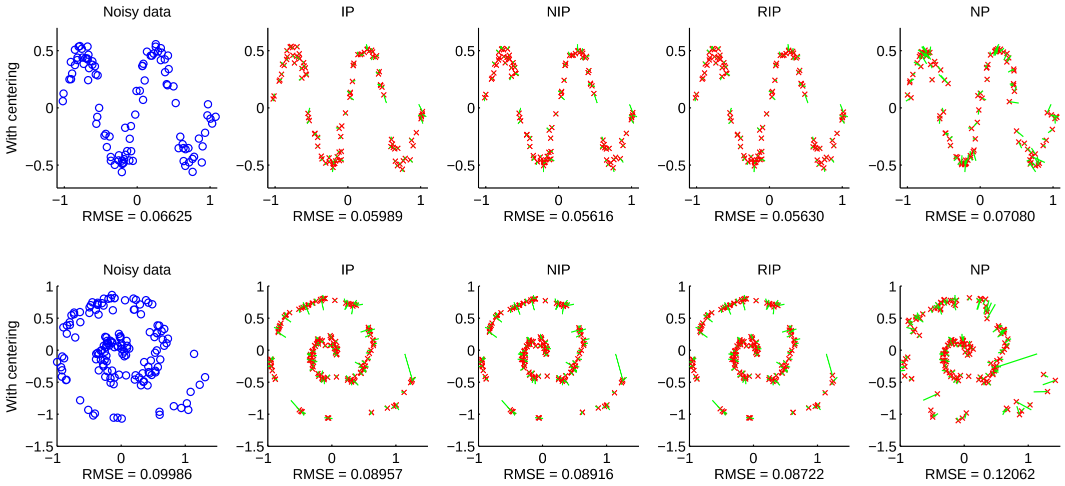


Figure 2.5: De-noising of the sine dataset with the variance $c = 0.1$ and $n = 15$ components for projection, and the spiral dataset with $c = 0.1$ and $n = 20$. For both cases centering is applied.

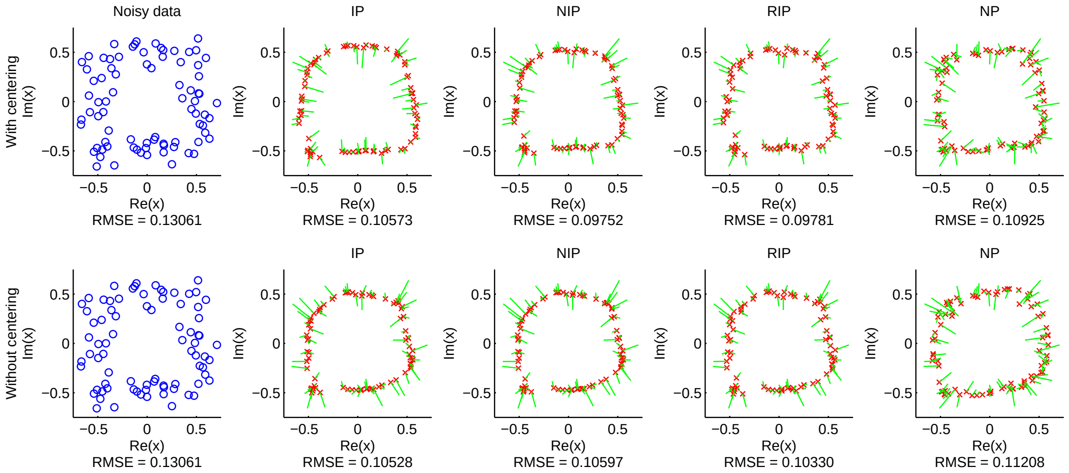


Figure 2.6: De-noising of the complex-valued square dataset with $c = 0.25$ and $n = 6$, with and without centering.

Chapter 3

Kernel PCA and Pre-Image Iterations for Speech Enhancement

PCA has been successfully applied for speech enhancement in the context of subspace methods [17, 40]. Kernel PCA – the non-linear extension of PCA – has first been applied to de-noise images [3, 6]. In speech processing, kernel PCA has been used to extract robust features from reverberant speech for ASR [25].

In this chapter, we introduce the application of kernel PCA for speech enhancement. We apply kernel PCA on feature vectors extracted from complex-valued spectral data. As we are interested in the enhanced signal in time domain, the pre-images of the processed samples have to be found. We first compare different pre-image methods and propose a combined method, which results in fewer artifacts in the time domain signal. We observed that the number of components n used in the projection step of kernel PCA and the weighting coefficients γ_i have only a minor influence on the result of the de-noising process. The experiments indicate that de-noising does not result from the projection operation but rather from the pre-image computation. These observations form the basis for our method called *pre-image iterations* for speech enhancement. PI constitute a simplification of an iterative pre-image method while the enhancement performance is maintained. De-noising of a feature vector is achieved by forming linear combinations of noisy feature vectors that are weighted by a kernel measuring the similarity between the currently enhanced vector and the other noisy feature vectors.

We show that pre-image iterations are related to non-local neighborhood filtering and to the non-local means algorithm which have been applied for de-noising in image processing [7, 27, 41, 42]. Recently, these methods have gained attention for speech enhancement, namely for the application of transient noise suppression in speech [43]. Finally, we use the knowledge derived from the convergence behavior of pre-image iterations for VAD in the spectral domain. This can be used to derived a mask for musical noise suppression in enhanced speech.

3.1 Time-Frequency Processing

Before describing the experimental framework in detail, we will provide a general discussion of time-frequency processing, which is often applied in audio and speech processing applications. By time-frequency processing, we mean that a two-dimensional representation is created from the time-domain signal, with one dimension corresponding to time and the other to frequency – the graphical representation of the magnitude is known as spectrogram. The time-frequency representation is modified according to the respective signal processing application, then the processed time-domain signal is reconstructed [44, 45, 46, 47].

Time-frequency processing is characterized by three steps: In the *analysis step*, the time-domain signal is transformed using a transformation such as the short-time Fourier transformation. In the *modification step*, modifications are applied to the resulting time-frequency representation. Finally, in the *synthesis step*, the inverse of the modified short-time Fourier transform (STFT) is computed and the time-domain signal is reconstructed from the resulting frames by either the *filter bank summation* method or by *overlap-add* (OLA) synthesis [44, 45, 2, 48]. In speech enhancement, typically the overlap-add synthesis is applied.

Let us consider a discrete-time signal $x[n]$, where n denotes the time index. In the analysis step, time segments – so-called frames – are extracted from the signal, windowed, and the discrete STFT is computed by application of

$$X[n,k] = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega_k m}, \tag{3.1}$$

where $\omega_k = \frac{2\pi k}{K}$, k is the index of the discrete frequencies, K is the number of Fourier coefficients (or frequency bins), and $w[n]$ is the analysis window [44, 2]. In speech processing, usually a Hamming window with a duration of 20-40 ms is applied [47].

In the modification step, the frequency representation is modified, i.e., the values of the frequency bins are changed, e.g., by a gain function. Mostly, modifications are only applied to the magnitude values and for synthesis the phase of the original signal is used.

In the synthesis step, the inverse short-time Fourier transformation is applied on the modified spectrum $\tilde{Y}[n,k]$ of each frame and the resulting frames are combined by overlap-add synthesis [44, 45, 49]. For an explanation of the overlap-add method, assume that in the analysis step the STFT is computed every R samples and that no modification is applied. Let us denote the STFTs by $X[rR,k]$ [2, 44]. The inverse discrete STFT for one time frame is

$$y_r[n] = \frac{1}{K}\sum_{k=0}^{K-1} X[rR,k]e^{j\omega_k n}, \tag{3.2}$$

which is equivalent to

$$y_r[n] = x[n]w[rR-n]. \tag{3.3}$$

The frames $y_r[n]$ are summed in an overlapping manner such that

$$y[n] = \sum_{r=-\infty}^{\infty} y_r[n] = x[n] \sum_{r=-\infty}^{\infty} w[rR-n]. \tag{3.4}$$

If the window is chosen such that the summation over the window in (3.4) equals a constant, the original signal $x[n]$ can be reconstructed perfectly up to the constant, such that

$$y[n] = C \cdot x[n], \tag{3.5}$$

assuming no modification in the modification step.

In [49], Griffin and Lim address the issue that the STFT is modified in many speech processing applications such as enhancement. Let us denote the modified continuous STFT by $\tilde{Y}(rR,\omega)$. An arbitrary $\tilde{Y}(rR,\omega)$ may not be a valid STFT, i.e., there is possibly no signal whose STFT is $\tilde{Y}(rR,\omega)$. For reconstruction we need to find a time-domain signal $y[n]$ that has an STFT $Y(rR,\omega)$ that approximates $\tilde{Y}(rR,\omega)$. Griffin and Lim propose to minimize the distance between $y[n]$ and $\tilde{Y}(rR,\omega)$

$$d(y[n],\tilde{Y}(rR,\omega)) = \sum_{r=-\infty}^{\infty} \frac{1}{2\pi}\int_{\omega=-\pi}^{\pi} |Y(rR,\omega)-\tilde{Y}(rR,\omega)|^2 d\omega, \tag{3.6}$$

which they define as squared error between $Y(rR,\omega)$ and $\tilde{Y}(rR,\omega)$ integrated over ω and summed over all r, where ω denotes the frequency and the STFT is computed every R samples as before. The distance d is written as function of $y[n]$ and $\tilde{Y}(rR,\omega)$ to make explicit that $Y(rR,\omega)$ is a valid STFT while this is not guaranteed for $\tilde{Y}(rR,\omega)$. Equation (3.6) can be rewritten by application of Parseval's theorem for the continuous STFT [45]

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi}\int_{-\pi}^{\pi} |X(\omega)|^2 d\omega, \tag{3.7}$$

such that

$$d(y[n],\tilde{Y}(rR,\omega)) = \sum_{r=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} [y_r[n]-\tilde{y}_r[n]]^2, \tag{3.8}$$

where $y_r[n] = y[n]w[rR-n]$ and $\tilde{y}_r[n] = \frac{1}{2\pi}\int_{\omega=-\pi}^{\pi}\tilde{Y}(rR,\omega)e^{j\omega n}d\omega$. Now, d can be minimized by setting the gradient with respect to $y[n]$ to zero

$$\nabla_{y[n]} d = 2\sum_{r=-\infty}^{\infty} \left(y[n]w[rR-n]-\tilde{y}_r[n]\right) w[rR-n] = 0. \tag{3.9}$$

This results in the closed form solution [49, 50]

$$y[n] = \frac{\sum_{r=-\infty}^{\infty} \tilde{y}_r[n]w[rR-n]}{\sum_{r=-\infty}^{\infty} w^2[rR-n]}. \tag{3.10}$$

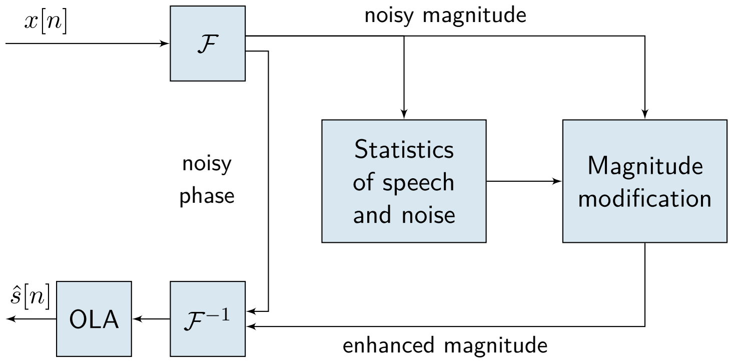


Figure 3.1: General processing chain of many speech enhancement algorithms. Processing is performed on the magnitude, for synthesis the phase of the noisy signal is used.

So the difference to the standard overlap-add method is that the inverse $\tilde{y}_r[n]$ of the modified STFT is windowed by a synthesis window before adding and that the values of the window $w[rR-n]$ are squared before the summation for normalization. The synthesis window is usually directly deduced from the analysis window. Similar to the standard overlap-add method, the window can be chosen such that the summation $\sum_{r=-\infty}^{\infty} w^2[rR-n]$ is unity for all n.

To get a more precise idea about the method, consider a signal $x[n]$ for which the STFT $X[rR,k]$ is computed using (3.1). The original time-domain signal without modification is recovered by application of the the inverse short-time Fourier transformation (3.2) and the result $y_r[n]$ is windowed with the synthesis window, which leads to $y_r[n]w[rR-n]$. For simplicity, we use the same window for analysis and synthesis. After summation and normalization in (3.10) the re-synthesized signal is

$$y[n] = \frac{\sum_{r=-\infty}^{\infty} y_r[n]w[rR-n]}{\sum_{r=-\infty}^{\infty} w^2[rR-n]} = \frac{\sum_{r=-\infty}^{\infty} x[n]w^2[rR-n]}{\sum_{r=-\infty}^{\infty} w^2[rR-n]} = x[n]. \tag{3.11}$$

Thus, the original signal is reconstructed if no modification is applied on the STFT $X[rR,k]$. Otherwise, if the STFT is modified, a signal is constructed corresponding to an STFT that optimally approximates $\tilde{Y}[rR,k]$ in the least squares sense [50].

3.2 General Structure of Speech Enhancement Methods

Now, let as take a look on the general structure of speech enhancement algorithms. Many speech enhancement algorithms perform enhancement by modifying the magnitude spectrum of the noisy signal while leaving the phase untouched. The phase is only used for the inverse transformation $\mathcal{F}^{-1}$ from frequency to time domain as illustrated in Figure 3.1. At low signal-to-noise ratios, this can reduce the speech quality [2].

To gain further insights, let us consider a speech signal $s[n]$ that is corrupted by additive noise $d[n]$. Then the noisy signal $x[n]$ is

$$x[n] = s[n] + d[n]. \tag{3.12}$$

By application of (3.1), the transform of the time-domain signal $x[n]$ is given as

$$X[n,k] = S[n,k] + D[n,k] \tag{3.13}$$

where $X[n,k]$, $S[n,k]$ and $D[n,k]$ are the STFTs of the noisy signal, the clean speech signal and the noise, respectively. Each STFT can be expressed in terms of magnitude $|X[n,k]|$ and phase θ_X, such that

$$X[n,k] = |X[n,k]|e^{j\theta_X} = |X[n,k]|(\cos\theta_X + j\sin\theta_X). \tag{3.14}$$

The phase of the noisy signal is determined by the phases of the clean signal and the noise. Let us denote the phases of the clean speech signal and the noise by θ_S and θ_D, respectively. The phase of the noisy signal θ_X can be derived as follows (for convenience we drop the indices n and k) [2]: First, we treat the real and imaginary parts of (3.13) separately, such that

$$|X|\cos\theta_X = |S|\cos\theta_S + |D|\cos\theta_D \tag{3.15}$$

$$|X|\sin\theta_X = |S|\sin\theta_S + |D|\sin\theta_D \tag{3.16}$$

Squaring (3.15) and (3.16) and adding the two equations together leads to

$$|X|^2 = |S|^2 + |D|^2 + 2|S||D|\cos(\theta_S - \theta_D). \tag{3.17}$$

In the case of speech enhancement the aim is to estimate the clean signal spectrum $S[n,k]$ from $X[n,k]$. The above equation shows, that the phase of the clean signal cannot be estimated exactly from the noisy signal as neither the phase of the clean signal θ_S nor the phase of the noise θ_D are known but only the difference between them.

3.3 System Overview of Kernel PCA and Pre-Image Iterations[1]

In contrast to most speech enhancement methods, we propose a method that processes complex-valued feature vectors extracted from the sequence of STFTs. Figure 3.2 illustrates the general structure. As the processing is based on the complex-valued spectral bins, we do not explicitly use the phase of the noisy signal for reconstruction of the time-domain signal. The handling of complex-valued data is facilitated by using kernel methods and especially the Gaussian kernel.

[1] This section is based on [51].

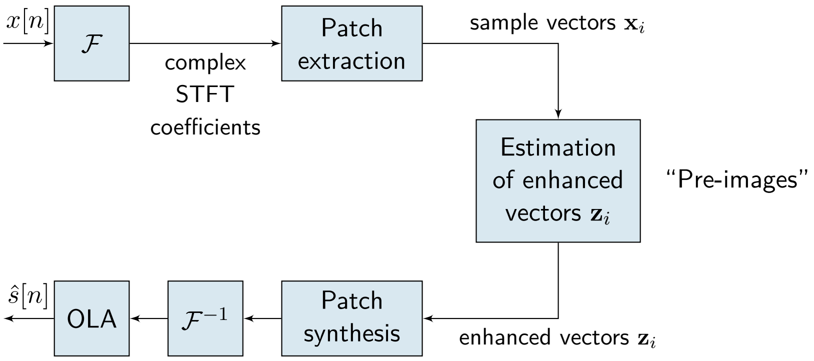


Figure 3.2: General system overview of our speech enhancement methods. Processing is performed on complex-valued STFT coefficients.

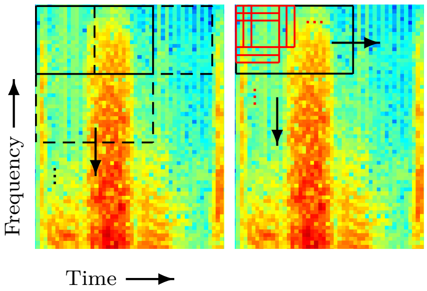


Figure 3.3: Left hand side: Extraction of frequency bands covering a time range of 10 patches and a frequency range of 8 patches (with 50% overlap along the time axis and no overlap along the frequency axis). Right hand side: Extraction of patches from one frequency band, where the patches cover 12 × 12 bins with an overlap of 10 bins in time and frequency. (Here shown on the clean signal for better visibility.)

For enhancement by kernel PCA and pre-image iterations we use the same feature extraction and synthesis. First the 256-point STFT is computed from frames of 16 ms. The frames have an overlap of 50% and a Hamming window is applied. The resulting time-frequency representation is split into time segments of 0.25 ms. Each segment is split on the frequency axis to reduce computational costs which results in so-called *frequency bands*. Sample vectors are retrieved from these frequency bands by first extracting quadratic patches in an overlapping manner, where the size of each patch is 12×12 with an overlap of 11. This is illustrated in Figure 3.3. On the left hand side, frequency bands are marked as black rectangles, on the right hand side, quadratic patches within one frequency band are marked as red squares. In previous experiments, windowing of the patches was beneficial, so a 2D Hamming window is applied. Then, the values in the patches are re-ordered in column-major order to form the sample vectors $\mathbf{x}_i \in \mathbb{C}^{144}$. The frequency bands cover a frequency range corresponding to 8 patches (i.e. 19 bins) and a time range corresponding to 20 patches (i.e. 31 bins). Along the frequency axis bands have an overlap of 50% or no overlap – depending on the experiment – and along the time axis the overlap is

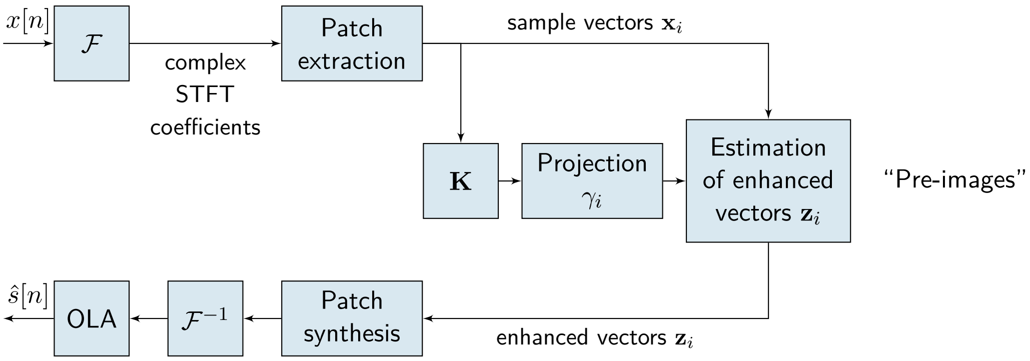

$x[n]$
$\mathcal{F}$
complex STFT coefficients
Patch extraction
sample vectors $\mathbf{x}_i$
$\mathbf{K}$
Projection γ_i
Estimation of enhanced vectors $\mathbf{z}_i$
"Pre-images"
$\hat{s}[n]$
OLA
$\mathcal{F}^{-1}$
Patch synthesis
enhanced vectors $\mathbf{z}_i$

Figure 3.4: Kernel PCA for speech enhancement.

10 patches. This configuration was chosen due to good empirical results.

After processing, the enhanced audio signal is synthesized by reshaping the enhanced sample vectors $\mathbf{z}_i$ to patches. The patches of all frequency bands belonging to one time segment are rearranged using the overlap-add method with weighting as described in [49], generalized for the 2D domain. Then, the STFT bins of overlapping time segments are averaged, the inverse Fourier transform is applied on the bins of each frame and the audio signal is synthesized with the weighted overlap-add method as in (3.10).

3.4 Kernel PCA for Speech Enhancement[2]

The application of kernel PCA for speech enhancement is illustrated by the block diagram in Figure 3.4. First, the STFT and the feature extraction as described in Section 3.3 are performed. One kernel matrix is built from the feature vectors of each frequency band. Each kernel matrix is centered according to (2.18), then the eigenvalue decomposition (2.15), normalization of the eigenvectors $\boldsymbol{\alpha}_k$ (2.16) and the projection of the data onto the eigenvectors $\mathbf{v}_k$ (2.17) are performed. A Gaussian kernel is used

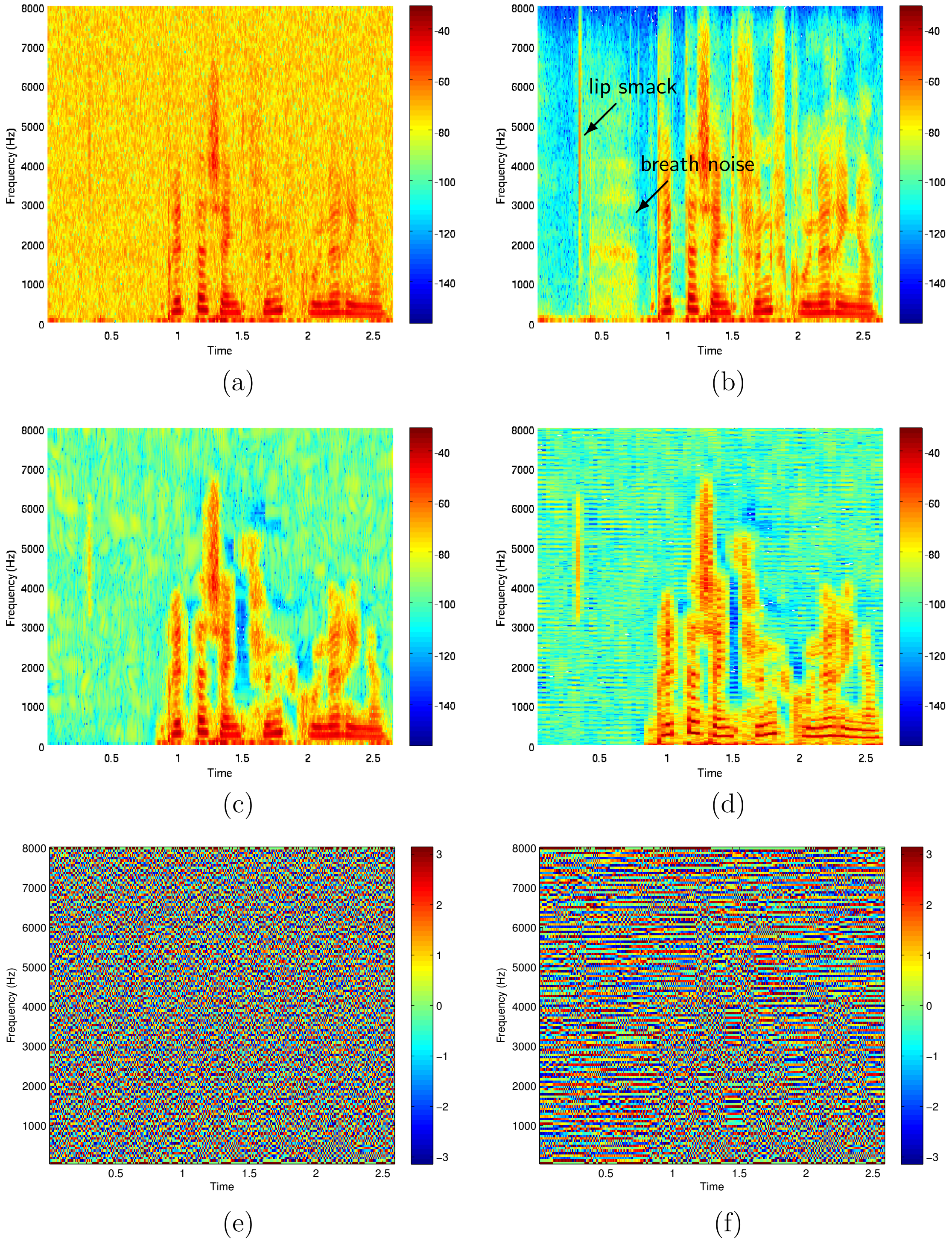
$$k(\mathbf{x}_i, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{c}\right). \tag{3.18}$$

The kernel variance c strongly influences the degree of de-noising. Therefore, the value is set by testing several values on a development set and choosing the one with the best performance. The pre-images, i.e., the enhanced sample vectors are computed iteratively using iterative pre-imaging with normalization (cf. (2.34)),

$$\mathbf{z}_j^{t+1} = \frac{\sum_{i=1}^{M} \tilde{\gamma}_i k(\mathbf{z}_j^t, \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{M} \tilde{\gamma}_i k(\mathbf{z}_j^t, \mathbf{x}_i)}, \tag{3.19}$$

where $\mathbf{z}_j^{t+1}$ is the j^{th} enhanced sample within a frequency band at iteration $t+1$, $\mathbf{x}_i$ are the noisy samples, $\tilde{\gamma}_i$ is given by (2.35) and M is the number of samples

[2] This section is based on [30].



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.5: The utterance "Britta schenkt fünf grüne Ringe." produced by a female speaker of the *airbone* database. Note that the beginning is free of speech but contains a lip smack and breath noise. Spectrogram of the (a) signal corrupted by additive white Gaussian noise at 10 dB SNR, (b) clean signal, (c) signal enhanced by the kernel PCA method, and (d) enhanced by kernel PCA and plotted with higher frequency resolution. (e) phase of the noisy signal, (f) phase after kernel PCA. The pattern visible in the phase plot (f) causes the harmonic artifacts in (d).

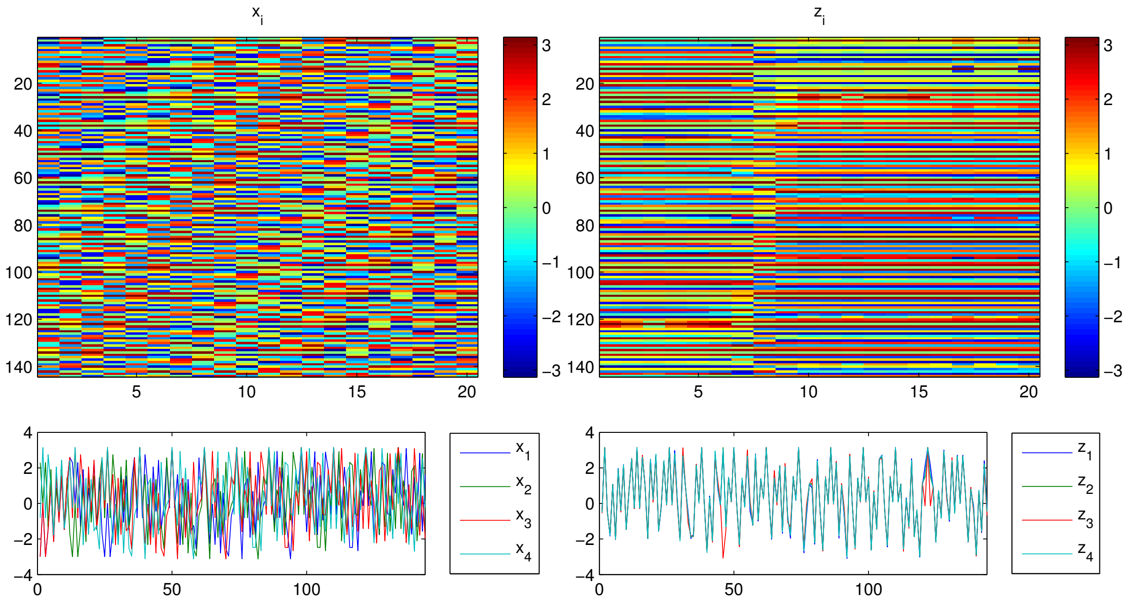


Figure 3.6: Phase of the feature vectors $\mathbf{x}_i$ before kernel PCA and $\mathbf{z}_i$ after kernel PCA for one frequency band, in the top left and top right plot, respectively. The matrix columns contain the feature vectors. The processing causes an alignment of the phase that induces the artifacts visible in Figure 3.5 (d). The bottom right plot illustrates the alignment of the phase after kernel PCA in comparison to the phase before kernel PCA (left) by showing the first four feature vectors.

in the frequency band. To enhance the sample vector $\mathbf{x}_j$ we initialize $\mathbf{z}_j^0$ with the noisy sample $\mathbf{x}_j$ and iterate (3.19) until convergence. Finally, the sample vectors are rearranged to patches and the audio signal is synthesized as described in Section 3.3.

Figure 3.5 shows the spectrograms of an utterance of the *airbone* database. The utterance is spoken by a female speaker and has been corrupted by additive white Gaussian noise (AWGN) at 10 dB SNR. Figure 3.5 (a) and (b) show the spectrograms of the corresponding noisy and clean signal, respectively. Figure 3.5 (c) shows the spectrogram after enhancement by the kernel PCA method using $n = 1$ component. Listening to the utterance reveals that noise is removed. No musical noise occurs, however, a buzz-like artifact is introduced. Looking at the spectrogram with a higher frequency resolution in Figure 3.5 (d) shows that the artifacts correspond to harmonics that smoothly change over time. The frequency of the artifact is related to the number of Fourier coefficients used for the STFT.

Further investigation revealed that these artifacts appear because different feature vectors converge to the same solution and their phase is aligned. Figure 3.5 (e) and (f) show the phase of the noisy signal and after enhancement, respectively. In some parts the plot in Figure 3.5 (f) shows a regular structure. It can be shown that the phase of the feature vectors shows an alternating pattern. After enhancement, the aligned feature vectors induce the harmonic structure visible in Figure 3.5 (d).

Figure 3.6 illustrates the phase of the samples vectors $\mathbf{x}_i$ and $\mathbf{z}_i$, i.e., before and after kernel PCA within one frequency band, in the left and right top plot, respectively. Each column of the plotted matrices represents one feature vector. The bottom plots in Figure 3.6 show the phase of the first four feature vectors. Before kernel PCA the phase is not aligned (left), while after processing it is aligned (right). This results in the changed phase values after enhancement. If the window length and the number of Fourier coefficients are changed, the pattern in the spectrogram persists but the frequencies change.

3.5 Kernel PCA with Combined Pre-Imaging[3]

The applied pre-image method strongly influences the outcome of the de-noising process. Therefore, similar to the experiments on synthetic data described in Section 2.2.2, we compare the same four pre-image methods for the application of speech enhancement. The following observations were made:

1. Iterative pre-imaging (IP) by Mika et al. [3] without normalization of γ_i often fails to converge and the audio signal is mostly zero.

2. Normalized iterative pre-imaging (NIP) proposed by Kwok and Tsang [33] is stable. The audio signal is de-noised, however a buzz-like artifact occurs that is related to the frame rate of the analysis window. This method is already discussed in Section 3.4.

3. Regularized iterative pre-imaging (RIP) by Abrahamsen and Hansen [34] results in good de-noising, depending on the value of η. The same artifact as for NIP appears.

4. Non-iterative pre-imaging (NP) introduced by Honeine and Richard [35] returns no meaningful audio signal. If the regularization parameter η is set to zero the audio signal contains similar artifacts as the signal from NIP while the speech signal is suppressed. When $\eta = 0$ the reconstruction of a pre-image is $\mathbf{z} = \mathbf{X}\boldsymbol{\gamma}$ (cf. (2.37). Consequently, each $\mathbf{z}$ is a linear combination of data samples $\mathbf{x}_i$ like for kernel PCA but with different weights. In the case of kernel PCA the speech signal is not attenuated. This is caused by the kernel weights $k(\mathbf{z}_j, \mathbf{x}_i)$ in the linear combination in (3.19). If a feature vector mainly contains speech components the kernel values between this feature vector and all the other feature vectors are close to zero while the kernel value between the feature vector and itself is one (see also the discussion in Section 3.6.1). Therefore, the pre-image is close to the original noisy feature vector. With the reconstruction of Honeine and Richard with $\eta = 0$, no kernel is used and the speech feature vectors are averaged, which leads to attenuation. In the case of feature vectors containing mostly noise the weighting of kernel PCA with NIP

[3] This section is based on [39].

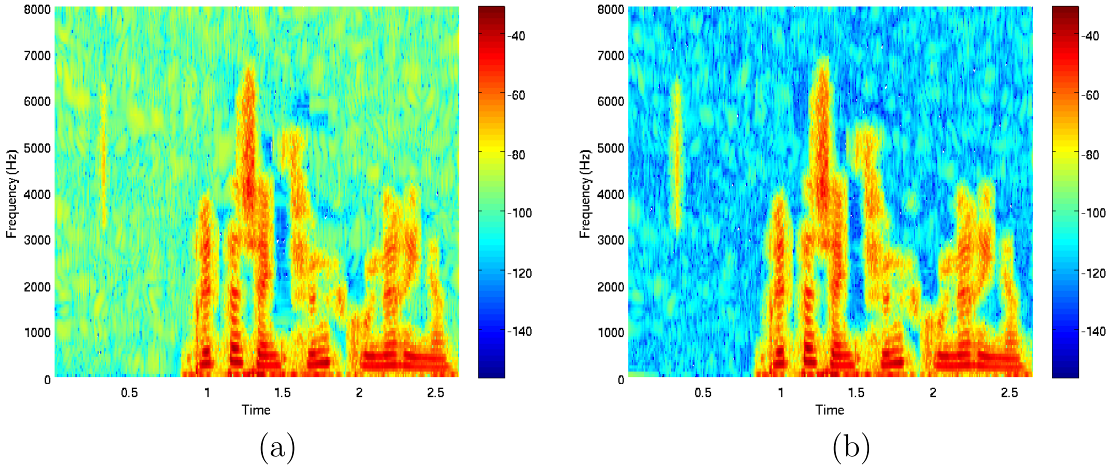


Figure 3.7: Spectrograms of the speech utterance in Figure 3.5 corrupted by white noise at 10 dB SNR and enhanced using (a) kPCA with NIP and (b) kPCA with combined pre-imaging (CO). The method with combined pre-imaging introduces considerably fewer artifacts.

and NP is similar as the kernel weights $k(\mathbf{z}_j, \mathbf{x}_i)$ in (3.19) are not dominated by one value. Therefore, the resulting signal mostly consists of the artifact while the speech is attenuated.

From these observations we deduced that the combination of NIP with NP could reduce the buzz-like artifact since NP with $\eta = 0$ basically only models the artifacts also occurring in NIP. Indeed, a subtraction of the signal of NP from the signal of NIP in time domain results in a signal of better quality as the buzz-like artifact is significantly reduced. A comparison of the spectrogram after enhancement by kernel PCA with NIP in Figure 3.7 (a) with the spectrogram after enhancement by the combined method in Figure 3.7 (b) shows the reduction of the artifact, while a reduction of the speech quality cannot be perceived. Listening to the enhanced utterances confirms better audio quality.

3.6 Pre-Image Iterations for Speech Enhancement[4]

When subspace methods are applied for speech enhancement, the number of components used for the projection step of PCA is a key parameter. In our framework with kernel PCA, we empirically observed that the number of components used for projection has only a minor, almost no, effect on the outcome of the de-noising process. The de-noising quality is rather the same whether projection is performed on one or more components. De-noising is primarily influenced by the kernel weights and by the value of the kernel variance. Therefore, we completely neglect the projection

[4] This section is based on [52].

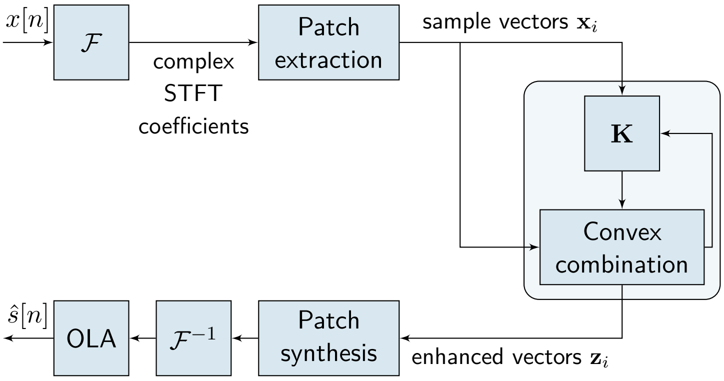


Figure 3.8: Pre-image iterations for speech enhancement.

coefficients $\tilde{\gamma}_i$ in (3.19) by setting them to one. We call this *pre-image iterations* for speech enhancement.

The pre-image in (3.19) is always a linear combination of the noisy input samples $\mathbf{x}_i$. In the case of kernel PCA, the weights of the linear combination are determined by the kernel $k(\cdot,\cdot)$ and the projection coefficients $\tilde{\gamma}_i$ (or γ_i if centering is omitted). With pre-image iterations, the linear combination only depends on the kernel. The Gaussian kernel serves as similarity measure between two samples. If the samples are equal, it is one, if they are very distinct, it is close to zero. The variance c is used as parameter to scale the degree to which samples are treated as similar.

The PI method is illustrated in the block diagram in Figure 3.8. The equation used for computation of an enhanced feature vector is

$$\mathbf{z}_j^{t+1} = \frac{\sum_{i=1}^{M} k(\mathbf{z}_j^t, \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{M} k(\mathbf{z}_j^t, \mathbf{x}_i)}. \tag{3.20}$$

It can be reformulated as

$$\mathbf{z}_j^{t+1} = \sum_{i=1}^{M} \tilde{k}(\mathbf{z}_j^t, \mathbf{x}_i)\mathbf{x}_i, \tag{3.21}$$

where

$$\tilde{k}(\mathbf{z}_j^t, \mathbf{x}_i) = \frac{k(\mathbf{z}_j^t, \mathbf{x}_i)}{\sum_{m=1}^{M} k(\mathbf{z}_j^t, \mathbf{x}_m)} \tag{3.22}$$

is the normalized kernel. As the kernel function can only take values between zero and one, $\tilde{k}(\cdot,\cdot)$ is also constrained to values within the interval $[0,1]$ and it is normalized such that $\sum_{i=1}^{M} \tilde{k}(\mathbf{z}_j^t, \mathbf{x}_i) = 1$. Due to these constraints, the pre-image $\mathbf{z}_j$ can be seen as a convex combination of the training samples $\mathbf{x}_i$ [53]. In other words the de-noised sample lies in a convex hull spanned by the noisy samples.

We further extended (3.20) with additional regularization similar as in [34] (cf. (2.36)), such that

$$\mathbf{z}_j^{t+1} = \frac{\frac{2}{c}\sum_{i=1}^{M} k(\mathbf{z}_j^t, \mathbf{x}_i)\mathbf{x}_i + \eta\mathbf{x}_j}{\frac{2}{c}\sum_{i=1}^{M} k(\mathbf{z}_j^t, \mathbf{x}_i) + \eta}, \tag{3.23}$$

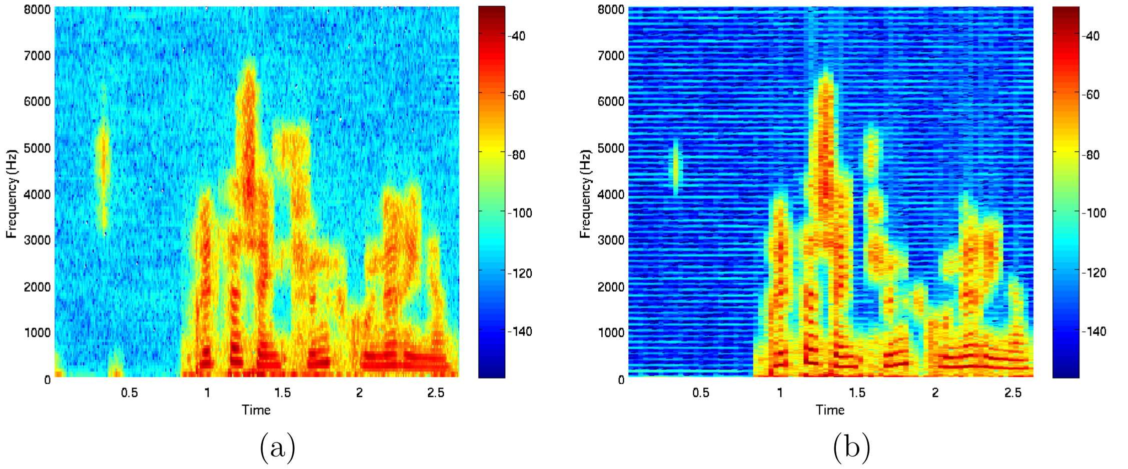


Figure 3.9: Spectrograms after enhancement by pre-image iterations with regularization plotted with (a) low and (b) high frequency resolution. Note that there is still a harmonic artifact, however, its magnitude is lower than in the case of kernel PCA and hence it cannot be perceived.

where $\mathbf{x}_j$ is the noisy sample, for which the pre-image should be found and $\eta \geq 0$ is the regularization parameter that determines the influence of the noisy sample $\mathbf{x}_j$ in the pre-image iterations.

The spectrogram of the pre-image iteration method with regularization in Figure 3.9 (a) shows that there are fewer artifacts compared to the kernel PCA method with NIP in Figure 3.7 (a). Figure 3.9 (b) shows the spectrogram with higher frequency resolution. It can be seen that there is still a harmonic artifact, however, its magnitude is considerably lower than in the case of kernel PCA. Listening to the utterance confirms that the artifact cannot be perceived. Compared to the method with combined pre-imaging (Figure 3.7 (b)), PI have a slightly higher low pass behavior and a little more residual noise is left. The enhanced signals of the pre-image iteration method and the method with combined pre-imaging sound very similar. With additional regularization in (3.23), the audio signal sounds similar as without regularization but with slightly more background noise that changes with the value of η. This result can be explained by equation (3.23). The different levels of background noise are caused by the weighting of the noisy samples by η in the regularization term.

In the next sections, we will first point out why pre-image iterations lead to de-noising and then discuss relations to other methods.

3.6.1 Analysis of Pre-Image Iterations

Pre-image iterations effect de-noising by a linear combination – or weighted average – of noisy feature vectors, where the weights are determined by the kernel. To

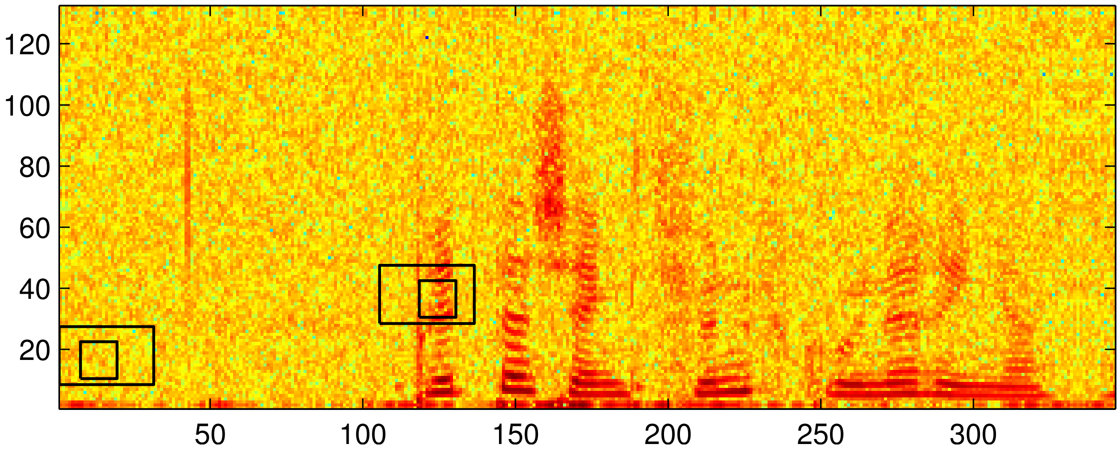


Figure 3.10: Magnitude of the spectral data for PI, shown for the example utterance corrupted by AWGN at 10 dB SNR. The frequency bands illustrated in Figure 3.11 and 3.12 are marked by boxes.

analyze the de-noising, we define the vector of kernel values

$$\mathbf{k}_j = [k(\mathbf{x}_j, \mathbf{x}_1), k(\mathbf{x}_j, \mathbf{x}_2), \ldots, k(\mathbf{x}_j, \mathbf{x}_M)]^T \tag{3.24}$$

computed between a feature vector $\mathbf{x}_j$ and all vectors $\{\mathbf{x}_i | i = 1, \ldots, M\}$ from one frequency band. This kernel vector always contains one large element, which equals one, because it denotes the similarity of the feature vector $\mathbf{x}_j$ with itself. The values of the other elements depend on the signal content in the examined feature vector. If the feature vector contains mostly speech, the other elements are only large if there are similar in-phase speech components within the frequency band, otherwise the other elements are close to zero. If the feature vector contains mostly noise, there are other elements larger than zero besides the element equal to one.

De-noising in the PI framework is therefore effected by the following means:

- Feature vectors containing in-phase speech components are combined, because the degree of similarity is high. Noise within these feature vectors is averaged out because it is randomly distributed. In practice and with the described configuration of the feature extraction, there are usually no in-phase feature vectors within a frequency band. Therefore, a feature vector containing speech components is only similar to itself and the noise reduction for this feature vector is limited. This is illustrated in Figure 3.11. The first and second column represent the noisy magnitude and the enhanced magnitude in a segment where speech is present. The third column shows a frequency band with speech components over several iterations. The marked patch (equivalent to a feature vector) and the corresponding kernel vector in the fourth column do not change in course of the iterations and no noise reduction is achieved for this patch.

- Feature vectors containing mostly noise do not exhibit a high degree of similarity (except to themselves), however, there is some similarity between all of

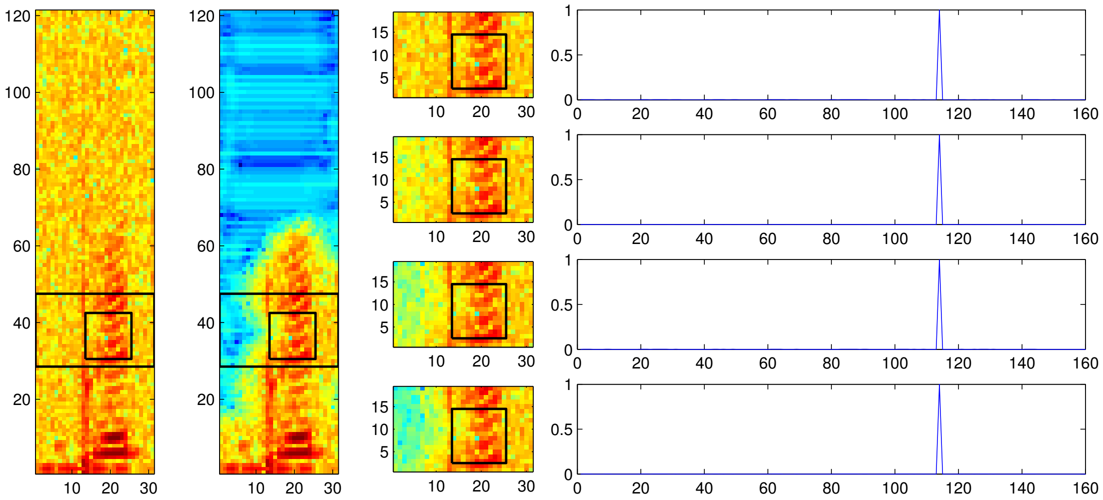


Figure 3.11: PI in a speech segment shown on the magnitude of the spectral data of the utterance in Figure 3.10. The columns show from left to right: (i) Noisy segment with one frequency band and a patch marked. The noise level is 10 dB SNR. (ii) Enhanced segment. (iii) The marked frequency band before de-noising and after one to three iterations. (iv) Kernel values between the marked patch and all other noisy patches in the band before de-noising and after one to three iterations. The kernel vector contains only one value significantly larger than zero and no averaging is performed for this patch. Note that the patches are extracted row-wise from left to right and from top to bottom.

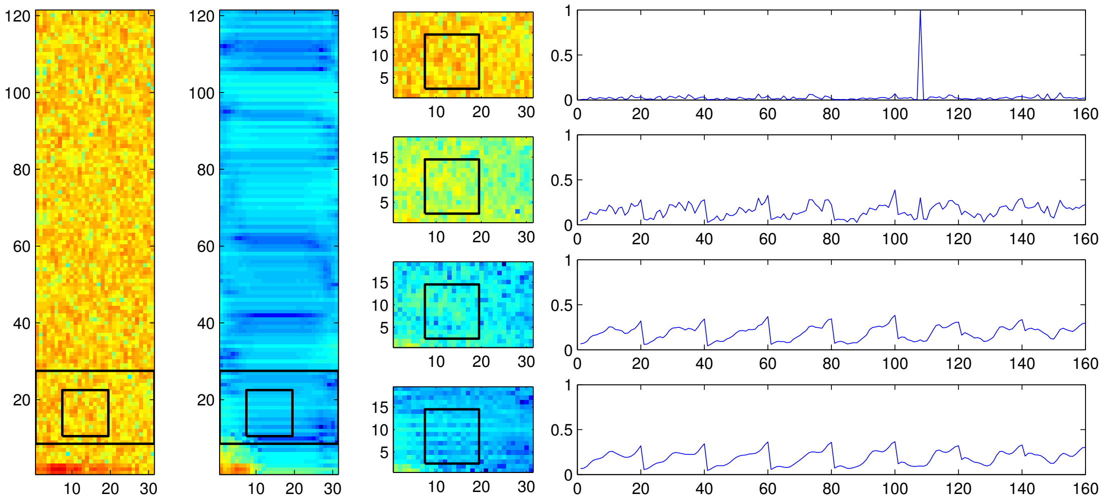


Figure 3.12: PI in a noisy segment. The columns show from left to right: (i) Noisy segment with one frequency band and a patch marked. (ii) Enhanced segment. (iii) The marked frequency band before de-noising and after one to three iterations. (iv) Kernel values between the marked patch and all other noisy patches in the band before de-noising and after one to three iterations. Note that in contrast to Figure 3.11, other kernel vector entries besides the entry equal to one are larger than zero and therefore contribute to the averaging in PI.

them. So, in contrast to feature vectors or patches where speech is present, there are other kernel values larger than zero, as illustrated in Figure 3.12. Consequently, feature vectors are averaged in the first iteration. In the next iteration, the kernel vector is computed between the enhanced feature vector and the noisy feature vectors. It turns out that the enhanced feature vector is even more similar to the noisy feature vectors than the original noisy feature vector. Therefore the kernel vector contains even larger elements and stronger averaging is performed. This is repeated until the weights are stable and convergence is reached. Note that the feature vectors are complex-valued and that the phase and magnitude are randomly distributed. Therefore the feature vectors add up destructively and the noise is canceled. If feature vectors based on the magnitude were averaged, the overall energy would not be reduced because magnitude vectors do not add up to zero as complex-valued feature vectors do.

One further observation can be explained by the behavior described above. Spectrograms of de-noised utterances show, that there is often noise left in between speech components, such as short speech pauses or generally around speech components. As patches containing speech are mostly similar to itself, no averaging is applied and there is no de-noising effect. To improve the de-noising ability, smaller patches could be beneficial. Furthermore, it could be useful to process longer frequency bands, as in this case the probability of finding in-phase patches similar to patches containing speech components would increase. In this case, however, the higher computational costs are a drawback.

3.6.2 Relation to the Soft k-Means Algorithm

Pre-image iterations are related to the soft k-means algorithm [54], which is used for clustering. The mean of one cluster is defined as

$$\mathbf{m}_k = \frac{\sum_{i=1}^{M} \frac{\exp(-\beta d(\mathbf{m}_k,\mathbf{x}_i))}{N_i}\mathbf{x}_i}{\sum_{i=1}^{M} \frac{\exp(-\beta d(\mathbf{m}_k,\mathbf{x}_i),)}{N_i}} \tag{3.25}$$

where

$$N_i = \sum_{l=1}^{K} \exp(-\beta d(\mathbf{m}_l, \mathbf{x}_i)), \tag{3.26}$$

$d(\mathbf{m}_l, \mathbf{x}_i)$ is the distance between the two points $\mathbf{m}_l$ and $\mathbf{x}_i$, β is the so-called stiffness parameter and K is the number of clusters. When the squared Euclidean distance is used, the exponential term is equivalent to the Gaussian kernel, where $c = 1/\beta$. So the soft k-means update of the cluster mean is the same as the update of the pre-image $\mathbf{z}_j$ apart from the normalization factor N_i (which is different for every $\mathbf{x}_i$).

3.6.3 Relation to Non-Local Neighborhood Filtering and to the Non-Local Means Algorithm

Performing de-noising on the time-frequency representation of speech incorporates some similarities to methods popular for image de-noising. Therefore, we give a short excursion on these image de-noising methods. However, they did not provide the basis for our methods.

Let us assume an image that is composed of pixels at the locations x_i that have an intensity value $y(x_i)$. In many approaches for image and signal de-noising the de-noised value $y^d(x_j)$ of the signal $y(x_j)$ at point x_j is based on the signal values $y(x_i)$ at points neighboring x_j. Gaussian or Gabor filters and anisotropic diffusion are examples for such de-noising approaches. Using these methods, image de-noising often is a trade-off between de-noising and distorting the image, e.g., blurring the image [7] – similar as speech enhancement often is a trade-off between de-noising and speech distortion.

Most of these methods, however, do not take into consideration one property of many signals and images, namely their *repetitive behavior*, which means that in most signals, patterns of the original noise-free signal occur at different time instances or spatial locations [27]. For time-domain signals this is the case for every periodic or nearly periodic signal, for instance neuronal spikes or heart beats. In images, there may as well be patches that occur at different spatial locations, e.g., in textures. For de-noising, it is preferable to exploit the occurrence of similar patterns in distant regions of the signal. Instead of using the values in the neighborhood of a point x_i, de-noising is performed over pixels belonging to similar patterns found anywhere in the signal. This gives directly rise to different meanings of *neighborhood*. Often, a neighborhood is understood as points that are spatially close. However, it can also be defined in terms of similarity of the pixels' intensity values. Yaroslavsky describes these different possibilities of defining a neighborhood of a pixel in [42].

Non-local neighborhood filtering and bilateral filtering are based on this notion of neighborhood. For a continuous one-dimensional signal, neighborhood filtering (NF) is defined as [27]

$$y_{NF}(x_j) = \frac{1}{D(x_j)} \int k(y(x_j), y(x_i)) y(x_i) dx_i, \tag{3.27}$$

where $y_{NF}(x_j)$ is the de-noised pixel, $k(\cdot, \cdot)$ denotes a kernel, for instance the Gaussian kernel, and

$$D(x_j) = \int k(y(x_j), y(x_i)) dx_i \tag{3.28}$$

is the normalization factor.

While neighborhood filtering as described above is based on the similarity between signal values at different locations, this approach can be extended to benefit also from spatial information. In this case, weighted averaging is performed over values where both the signal value $y(x_i)$ is close to $y(x_j)$ and the location x_i is close to x_j. This is also known as *bilateral filtering* [27, 55]. One popular example is the

Yaroslavsky filter that considers the spatial distance by only using pixels in the neighborhood B_ρ, which is defined as a ball with center at x_j and radius ρ. The filter can be formulated as [27, 7]

$$y_{YNF_\rho}(x_j) = \frac{1}{D(x_j)} \int_{B_\rho(x_j)} k(y(x_j), y(x_i))y(x_i)d(x_i) \tag{3.29}$$

where $D(x_j) = \int_{B_\rho(x_j)} k(y(x_j), y(x_i))d(x_i)$ is the normalization factor.

Iteration of Non-Local Neighborhood Filters

The simple application of (3.27) is often not sufficient to achieve de-noising [27]. This can be improved by the iterative execution of non-local filtering. For simplicity, let us consider a one-dimensional signal. For a discrete signal $y(x_i)$ the above equation changes to

$$y^d(x_j) = \frac{1}{D(x_j)} \sum_{i=1}^{N} k(y(x_j), y(x_i))y(x_i), \tag{3.30}$$

where $y^d(x_j)$ is the de-noised signal, $D(x_j) = \sum_{i=1}^{N} k(y(x_j), y(x_i))$ denotes the normalization factor. and $k(\cdot, \cdot)$ is a kernel.

For iterative execution, the above equation can be rewritten as

$$y_{t+1}(x_j) = \frac{1}{D_t(x_j)} \sum_{i=1}^{N} k(y_t(x_j), y_t(x_i))y_t(x_i), \tag{3.31}$$

where t denotes the iteration index, $y_0(x_j) = y(x_j)$ is the original noisy sample at $t = 0$ and $D_t(x_j) = \sum_{i=1}^{N} k(y_t(x_j), y_t(x_i))$ (for convenience we drop the superscript d).

In detail, three possible ways to iterate Equation (3.31) are described in [27]. The first way is to update the signal value, the kernel and the normalization factor according to the signal value in the previous iteration, as given in (3.31). The second way is to keep the kernel and the normalization factor fixed and only update the signal value

$$y_{t+1}(x_j) = \frac{1}{D_0(x_j)} \sum_{i=1}^{N} k(y_0(x_j), y_0(x_i))y_t(x_i), \tag{3.32}$$

where y_{t+1} is the de-noised signal value at iteration $t + 1$. The third possibility is to keep the signal value for weighted averaging fixed to the original noisy value and update the kernel value and the normalization factor, such that

$$y_{t+1}(x_j) = \frac{1}{D_t(x_j)} \sum_{i=1}^{N} k(y_t(x_j), y_t(x_i))y_0(x_i). \tag{3.33}$$

Pre-image iterations apply a slightly different scheme, where only the currently de-noised patch is updated. For a single signal value this iteration scheme is

$$y_{t+1}(x_j) = \frac{1}{\tilde{D}_t(x_j)} \sum_{i=1}^{N} k(y_t(x_j), y_0(x_i))y_0(x_i), \tag{3.34}$$

where the normalization factor is $\tilde{D}_t(x_j) = \sum_{i=1}^{N} k(y_t(x_j), y_0(x_i))$.

Non-Local Means

Buades [7] proposed the non-local means (NL) algorithm, which is inspired by non-local filtering. We comared PI to the block-wise implementation of the NL algorithm. Consider a discrete noisy image $y = \{y(x_i)|x_i \in I\}$ where $y(x_i)$ is the intensity value of the pixel at location x_i, I is a 2D grid of pixels and $\{x_1, \ldots, x_n\}$ forms a subset of I. Now, for each x_j define a neighborhood $W_j = x_j + B$ that is centered in x_j. B defines the size and the shape of the neighborhood. We now assume that all W_j form a connected subset of I, where intersections between the neighborhoods are allowed.

Then, for each W_j the NL algorithm evaluates to

$$y_{NL}(W_j) = \frac{1}{C_j} \sum_{x_i \in I} y(x_i + B) e^{-\frac{\|y(x_j+B)-y(x_i+B)\|_2^2}{h^2}}, \tag{3.35}$$

where $C_j = \sum_{x_i \in I} e^{-\frac{\|y(x_j+B)-y(x_i+B)\|_2^2}{h^2}}$, h^2 is a filtering parameter equivalent to the kernel variance in pre-image iterations and $y(x_i + B)$ denotes all intensity values of the neighborhood W_i centered at x_i. As overlapping neighborhoods are allowed, one pixel is generally assigned to several neighborhoods and the NL algorithm computes for each pixel one different value per neighborhood. To find the final value of a de-noised pixel at location x_i, the values corresponding to that pixel in different neighborhoods are averaged, such that

$$y_{NL}(x_i) = \frac{1}{|A_i|} \sum_{j \in A_i} y_{NL}(W_j)(x_i) \tag{3.36}$$

where $A_i = \{j|x_i \in W_j\}$ indicates all neighborhoods containing x_i and $y_{NL}(W_j)(x_i)$ denotes the value at x_i in the neighborhood W_j after application of (3.35).

To make the connection to the pre-image iteration method explicit, we reformulate Equation (3.35) in vector notation and insert the kernel function

$$\mathbf{w}_k = \frac{\sum_{i \in I} k(\mathbf{w}_j, \mathbf{w}_i)\mathbf{w}_i}{\sum_{i \in I} k(\mathbf{w}_j, \mathbf{w}_i)}. \tag{3.37}$$

where $\mathbf{w}_j$ is the vector resulting from reordering the elements of W_j in column major order and $k(\cdot, \cdot)$ is the Gaussian kernel. This is equivalent to the first iteration of the pre-image iteration equation (3.20)

$$\mathbf{z}_j^{t+1} = \frac{\sum_{i=1}^{M} k(\mathbf{z}_j^t, \mathbf{x}_i)\mathbf{x}_i}{\sum_{i=1}^{M} k(\mathbf{z}_j^t, \mathbf{x}_i)}$$

if the neighborhoods are equivalently chosen and if the search region for neighborhoods is reduced from the whole image to a sub-region of the image. One image sub-region then corresponds to a frequency band and a neighborhood corresponds to a patch of the pre-image iteration method. A substantial difference, however, is that in the case of speech enhancement the frequency bins – which correspond to the pixels – are complex-valued. Furthermore, for speech enhancement the patches are weighted by a 2D Hamming window. In the block-wise implementation of the non-local means algorithm no weighting is applied. Note, however, that in the original algorithm proposition a weighted Euclidean distance is used to assess the similarity between neighborhoods.

3.6.4 Relation to Diffusion Filters for Speech Enhancement

We have now shown the similarities between neighborhood filtering, non-local means, and the proposed pre-image iteration algorithm. It is interesting to note, that a connection can be established between neighborhood filtering and diffusion equations. In [55], Barash demonstrates first the relation between adaptive smoothing and anisotropic diffusion and then the relation between adaptive smoothing and bilateral filtering, i.e., neighborhood filtering, in image processing. In [27], the connection between neighborhood filtering and diffusion processes is analyzed in more detail.

In speech enhancement, non-local filtering techniques have only been adopted recently. In [28, 43], non-local neighborhood filters are employed to suppress transient noise. Transient noise consists of short bursts that most speech enhancement algorithms fail to suppress as they are restricted to stationary noise. The repetitive structure of transient noise that causes other enhancement algorithms to be unsuitable for suppression can be exploited by application of non-local filtering. Talmon et al. [43] note that the non-local neighborhood filter in their application is equivalent to non-local diffusion filters as described in [27]. Therefore, we further refer to their work as non-local diffusion filters (NLDF) .

They use a variant of the optimally modified log-spectral amplitude (OM-LSA) estimator [56] that takes into consideration both a noise estimate for transient noise and a noise estimate for stationary noise. The transient noise is estimated using a non-local filter, that averages similar occurrences of transient noise bursts. Non-local filtering is applied on the STFTs with normalized Gaussian kernels as weights

$$\tilde{k}(\boldsymbol{\phi}(n), \boldsymbol{\phi}(m)) = \frac{k(\boldsymbol{\phi}(n), \boldsymbol{\phi}(m))}{\sum_{i=1}^{M} k(\boldsymbol{\phi}(n), \boldsymbol{\phi}(i))}, \tag{3.38}$$

where $\boldsymbol{\phi}(n)$ is the short-time power spectral density (PSD) in time frame n computed by smoothing periodograms over time frames and $k(\cdot, \cdot)$ is the Gaussian kernel. Note that the normalized kernel $\tilde{k}(\cdot, \cdot)$ can be interpreted as a transition probability in a graph. Hence, Talmon et al. formulate the non-local filtering as iterations on a graph with STFTs as nodes and normalized kernels as weights.

To draw a relation between the application of non-local diffusion filters for transient noise reduction and pre-image iterations, we shortly provide an overview on differences and commonalities:

- **Features** For NLDF the STFTs of each frame are used as feature vectors. For PI quadratic patches are extracted from the sequence of STFTs and vectorized.

- **Kernel** Both methods employ a Gaussian kernel. In the case of NLDF the kernel is computed between short time power spectral densities and it is normalized, while in the case of PI the kernel is evaluated on the complex-valued feature vectors.
- **Phase** The usage of the PSD for the kernel computation in NLDF implies that the phase information is neglected when the similarity between frames is measured. This proceeding is intentional, as the phase depends on the location of a transient within the frame and a transient should be identified regardless of the exact location within the frame. However, the phase is taken into consideration when the OM-LSA estimator is applied. Pre-image iterations use the phase as a consequence of processing complex-valued Fourier coefficients.
- **Effects** The two algorithms have different intentions and therefore considerably different effects. NLDF employ non-local filtering for constructive averaging of transients, that results in a robust estimate of the PSD of the transient noise, which is further used for de-noising. Pre-image iterations aim at reducing stationary noise by relying on the assumption that noise is randomly distributed and averaging between feature vectors is consequently destructive, i.e., the noisy feature vectors cancel each other.

Thus, although NLDF and PI are related via non-local filtering, the purpose of the methods is different.

3.6.5 Role of the Phase

In this section, we provide a discussion on the role of the phase in speech enhancement. Many speech enhancement algorithms perform processing on the magnitude and use the phase of the noisy signal for reconstruction of the enhanced time-domain signal. The authors of [57] argue that the phase is of minor importance while in [47] its importance is highlighted. Besides speech enhancement, the estimation of the phase has also been subject of recent investigations in the field of single-channel source separation [58, 59]. We first discuss some experimental results reported in literature and then present experiments realized with pre-image iterations.

Generally, when reconstructing a signal from its Fourier representation, the phase plays an important role. In [57], Oppenheim and Lim discuss this issue and demonstrate examples from image and speech processing. It is shown that an image or a speech signal, that is reconstructed by using only the phase, contains more identifiable features than a signal that is reconstructed from the magnitude only. For speech, a signal reconstructed from the phase of a long segment of speech and unity magnitude is reported to preserve a high degree of intelligibility. Hence, for both speech and image signals the phase seems to cover much of the relevant information. Under certain assumptions about the signal, it is even possible to construct the complete signal – magnitude and phase – up to a scaling factor from the phase only (for details and examples see [57]). However, Oppenheim and Lim note that the importance of a the long-time phase does not necessarily imply that the short-time phase is of equal importance.

In [60], Wang and Lim discuss the *unimportance of phase in speech enhancement.* They start from the assumption that the short-time phase is relatively unimportant in speech enhancement relying on short-time analysis. They also note that the ear does not express any preference among either changes in the phase of a sinusoidal signal or changes in the relative phase of sinusoidal components of a signal. However, they remark that rapid fluctuations of the relative phases of sinusoidal components in speech are reported to cause considerable degradation of the speech quality.

In order to investigate the role of the phase in the field of speech enhancement, Wang and Lim conducted a subjective listening test. In short, they corrupted signals with AWGN at several SNRs in time-domain and created new signals by combining the magnitude and the phase obtained with different SNRs. Pairs of these mixed signals and the original noise-corrupted signals were presented to listeners who had to vote which signal had better quality. Through the presentation of the original signals with different SNR levels, Wang and Lim derived an equivalent SNR for which the signals with mixed magnitude and phase were preferred 50% of the time and the original noisy speech was preferred 50% of the time. The equivalent SNR was then used to determine the relative importance of the magnitude and the phase for speech enhancement.

The results in [60], lead to the conclusion that a more accurate phase estimate only results in higher equivalent SNRs for a limited number of conditions. In particular, this is the case when a long frame length for the Fourier transform is used (4096 samples at a sampling rate of 10 kHz) and at the same time the SNR is very low. Low SNRs are, however, the case where an accurate phase estimation will be difficult anyway, if it is estimated in addition to the magnitude. Further experiments showed that a decrease in the accuracy of the phase estimate can cause a considerable decrease in the equivalent SNR. Due to the difficulty of estimating the phase in low SNR conditions and the risk of degrading the signal be erroneous estimation, phase estimation is not recommended by Wang and Lim in low SNR scenarios. In a further experiment, the magnitude of the combined signal was replaced by a magnitude estimated using spectral subtraction. This, however, did not lead to a considerable improvement of the equivalent SNR.

The following conclusions are drawn from the experimental results in the framework of Wang and Lim: Estimating the phase independently from the magnitude might only slightly improve the quality and is therefore probably not worth the effort. On the other hand, an approach jointly based on phase and magnitude estimation might benefit from a more accurate phase estimation.

In [47], Paliwal et al. conducted a series of experiments, which were evaluated objectively by using the PESQ measure (see Section 4.3.1 for details on the PESQ measure) and subjectively by listening tests. In their experiments, they reproduced the results of Wang and Lim. However, with a different setup for the Fourier transformation they could show that processing the phase can improve the performance assessed by both objective and subjective evaluation.

Clean	clean signals
Noisy	noisy signals
MMSE	signals with the magnitude enhanced by the MMSE STSA method
MMSE-Matched-O	signals with the magnitude enhanced by the MMSE STSA method, matched analysis windows, and clean phase
MMSE-Mismachted-O	signals with enhanced magnitude, mismatched windows and clean phase
MMSE-Mismachted-N	signals with enhanced magnitude, mismatched windows and noisy phase
PSC	noisy signals with phase estimation
MMSE-PSC	enhanced signals with phase estimation

Table 3.1: Signal combinations evaluated in the study by Paliwal et al. [47].

To be more precise, in a first experiment they combined the noisy magnitude with the phase of the clean signal and evaluated the performance after inverse transformation and overlap add – this is called the oracle experiment. One set of utterances was created by using the same setup as Wang an Lim, namely by application of a Hamming window with 50% in the analysis step. These signal set is denoted by *Wang-O*. Another set of utterances was produced by using a different setup, again by application of a Hamming window but with 82.5% overlap and additional zero padding such that the number of Fourier coefficients was doubled. The same window was used for magnitude and phase computation, hence it is termed *Matched-O*. A third set of signals was created by using different windows for the magnitude and the phase computation – this is termed the mismatched case (*Mismatched-O*). Namely, a Hamming window was used for computation of the magnitude and a Chebyshev window was used for computation of the phase. For the utterances created by the Wang-O procedure, no significant improvement by using the phase of the clean signal could be explored. However, for the Matched-O approach the PESQ measure returned higher scores and the results in the listening test were superior as well. The signals created by the Mismatched-O procedure even attained significantly higher results than the other methods. This is explained by the fact that noise can be reduced by appropriately choosing the dynamic range[5] of the Chebyshev window, as the dynamic range controls the tradeoff between preservation of fine spectral detail and spectral smoothing.

In two further experiments, signals created with knowledge of only the noisy phase and signals with additionally enhanced magnitude were compared. In the experiments based on the noisy phase, the signals created using the mismatched analysis windows showed higher performance than the noisy signal, both by objective

[5] Attenuation of the side lobes or highest side lobe with respect to the main lobe, depending on the used definition.

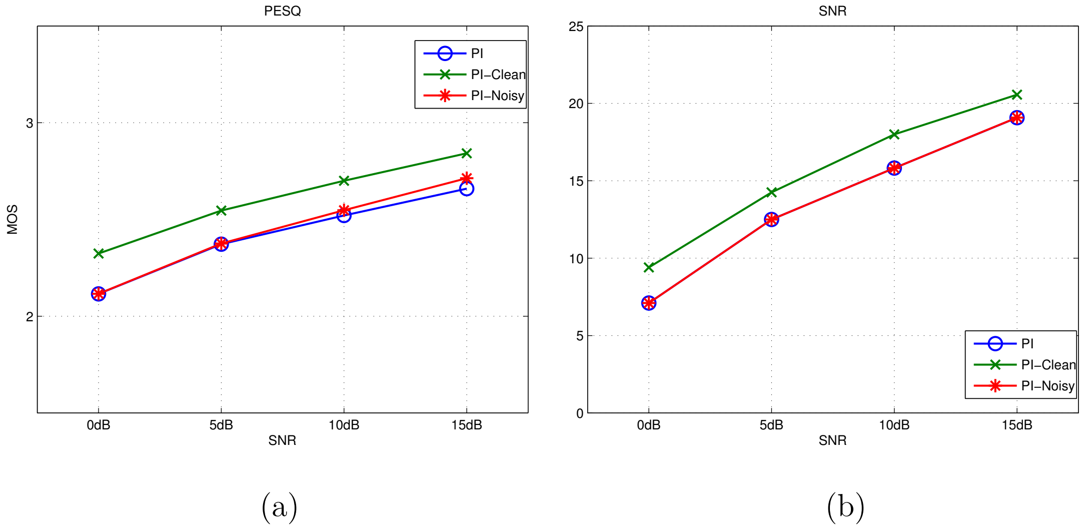
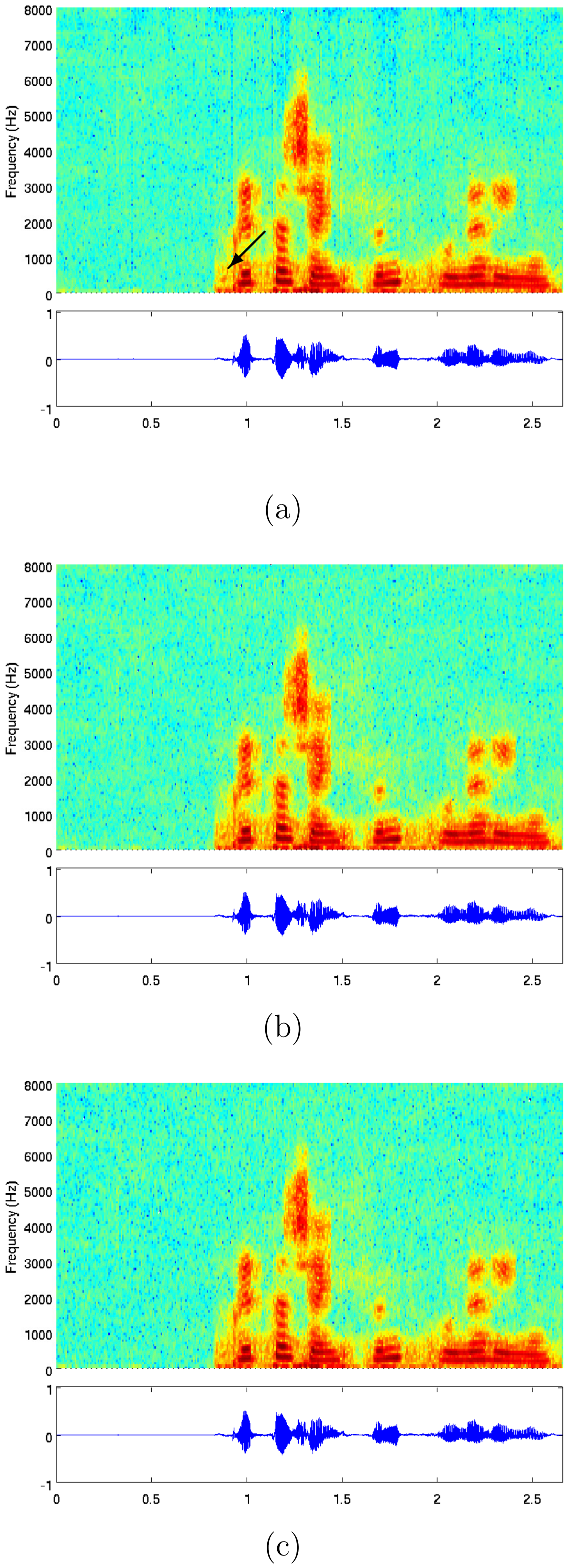


Figure 3.13: Comparison of the performance of signals after enhancement by pre-image iterations (PI) with signals obtained by combination of the magnitude of PI and the clean phase (PI-Clean) and the magnitude of PI and the noisy phase (PI-Noisy) in terms of (a) the PESQ measure and (b) the global SNR (SNR) for signals corrupted by AWGN at 0, 5, 10, and 15 dB SNR.

and subjective evaluation. Thus, similar to the oracle experiment, noise reduction can be achieved by the sole application of windows with different dynamic range for computation of magnitude and phase. If a low dynamic range is chosen for phase computation this apparently leads to noise reduction as the fine spectral detail is lost due to smoothing.

For the experiments including enhancement of the magnitude, the MMSE STSA estimator [12] was applied. Additionally, signals where the phase was estimated using the phase spectrum compensation (PSC) method [61, 62] were used for comparison. In total, eight types of signals as listed in Table 3.1 were evaluated. The main conclusions from these experiments are: The signals obtained by using the clean phase (MMSE-Matched-O and MMSE-Mismatched-O) perform best in objective evaluation. The signals created by enhancing the magnitude and in addition the phase by PSC achieve the next best score. The results of the subjective evaluation are slightly different. For this case, the score for MMSE-PSC is similar to the score for MMSE-Matched-O. This performance difference is explained as follows: The MMSE-Matched-O signal contains more spectral details while the MMSE-PSC signal provides better noise reduction. While the PESQ measure seems to focus on the spectral details, human listeners prefer a stronger noise reduction. In summary, these experiments suggest that processing the phase is promising to further improve the quality of enhanced speech.

To gain more insights about the role of the phase in PI, we compared the signals obtained by PI to signals based on the magnitude after PI and the phase of (i) the clean and (ii) the noisy signal. Note that, although we use the magnitude after PI, PI are still applied on the complex-valued feature vectors. We used the speech



(a)

(b)

(c)

Figure 3.14: Spectrograms of the sample utterance corrupted by AWGN at 5 dB SNR and (a) obtained by combination of the magnitude of PI with the clean phase, (b) obtained by combination of the magnitude of PI with the noisy phase, (c) after PI. The arrow in (a) indicates slightly stronger de-noising.

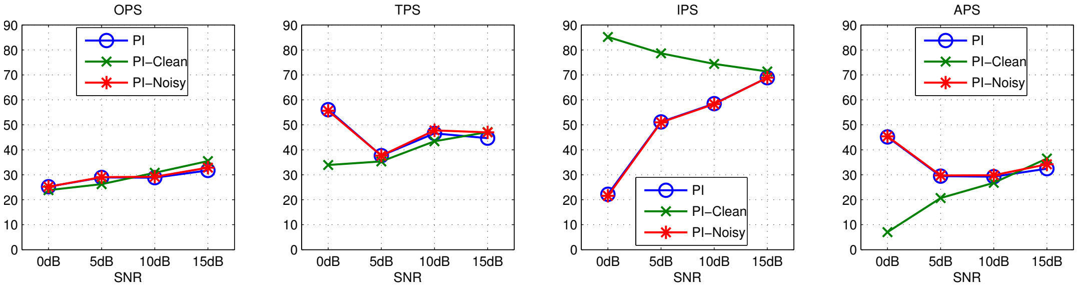


Figure 3.15: Comparison of the performance of signals after enhancement by pre-image iterations (PI) with signals obtained by combination of the magnitude after PI and the clean phase (PI-Clean) and the magnitude after PI and the noisy phase (PI-Noisy) for signals corrupted by AWGN at 0, 5, 10, and 15 dB SNR in terms of overall perceptual score (OPS), target perceptual score(TPS), interference perceptual score (IPS), and artifact perceptual score (APS).

utterances of the *airbone* database (see Section 4.1.1) with additive white Gaussian noise at four different SNRs, namely 0, 5, 10, and 15 dB. For evaluation we employed the PESQ measure. The results are presented in Figure 3.13 (a). The PESQ measure shows that the signals based on the clean phase attain higher scores than the other signals, thus the clean phase leads to an improvement. The scores achieved with the PI method are almost equal to the scores achieved by combination with the phase of the noisy signal. From these results, we conclude that handling the phase implicitly in PI has no disadvantage in comparison to using the phase of the noisy signal. However, as there is also no improvement an explicit phase estimation might be beneficial. Furthermore, it has to be considered that the results are influenced by the choice of the frame length, frame overlap and STFT size [47]. This is subject to future investigations.

Figure 3.14 shows the spectrograms of an utterance corrupted by white noise at 5 dB SNR and enhanced by PI, where (a) is created using the magnitude of PI and the clean phase, (b) is created using the magnitude of PI and the noisy phase and (c) is the signal obtained by the application of PI. A comparison of plot (a) and (b) reveals that noise in the signal with the clean phase is slightly better attenuated, as for instance in the region marked by the arrow. The computation of the global SNR confirms this impression, as can be seen in Figure 3.13 (b).

Listening to the utterance plotted in Figure 3.14 leads to further observations. First, the recording reconstructed with the clean phase contains a “crackling” artifact in the background. The recording based on the noisy phase and the recording obtained by PI are not affected by such an artifact. In all recordings, there is residual noise present in the background and the high frequency components are damped. The speech attenuation is a consequence of the lower energy in high frequency bands and of the large value for the kernel variance, which is necessary to achieve de-noising. Interestingly, the PESQ measure does not reflect the distortions that can clearly be perceived when listening to the signal with the clean phase.

The individual scores for the investigated utterance show the same tendency as the overall scores given in Figure 3.13.

Figure 3.15 shows the scores achieved by the PEASS measures [63]. The PEASS measures comprise the overall perceptual score (OPS), target perceptual score (TPS), interference perceptual score (IPS), and artifact perceptual score (APS) (see Section 4.3.3 for details on the PEASS measures). It is interesting to note that the APS is lower for the signal with the clean phase in the lower noise conditions, while the IPS is very high.[6] These tendencies are consistent with the impressions from listening. Furthermore, the OPS of the signal with clean phase is slightly lower than for the other methods at 0 and 5 dB SNR. This also agrees with the impression from listening, as for these SNR conditions the crackling artifact is perceivable. For the higher SNR cases this is a minor problem. As the PESQ and PEASS quality measures do not agree a systematic subjective evaluation is necessary to gain certainty about the subjectively perceived quality. This is, however, beyond the scope of this thesis.

3.7 Pre-Image Iterations with Determination of the Kernel Variance[7]

The kernel variance c is the key parameter to achieve good noise attenuation with PI. In the first pre-image iteration experiments [52], the SNR was assumed to be known and a suitable value for c was chosen according to the de-noising performance on a development dataset.

For the utterances of the *airbone* database there is some variation of the noise level in the same SNR condition. Therefore, it is difficult to choose one value for c that is optimal for all utterances in one SNR condition. Thus, we investigated how an optimal value of c can be chosen for each utterance based on an estimate of the noise level. For the determination of a suitable value of c we estimate the noise at the beginning of the utterance, assuming that there is no speech and stationary noise.

The power of a finite zero-mean signal $x[n]$ is equivalent to the signal variance and estimated using

$$\sigma^2 = \frac{1}{N} \sum_{n=0}^{N-1} |x[n]|^2, \tag{3.39}$$

where N is the length of the frame used for estimation. We use the signal power at the beginning of the noisy speech signal as noise estimate. To find a good setting for the frame length, we computed the noise estimates for frame lengths of 256, 512, 1024, and 2048 samples for the *airbone* database and compared them with the measured noise power of the entire recordings. The noise estimate with a frame length of 2048 has the lowest deviation from the power measured for the entire signal. With a sampling rate of 16 kHz, 2048 samples correspond to 128 ms. For

[6] Higher scores always denote better performance.

[7] This section is based on [51].

this time frame the assumption that there is no speech present holds for the *airbone* database.

We propose two variants of PI depending on the noise type. In the case of white noise, a mapping function is learned that maps the noise power to an appropriate value of c. In the case of colored noise, the noise power is not uniformly distributed over the frequency range. A single value for c therefore does not result in optimal de-noising. To account for this, c is determined separately for each frequency band from feature extraction. The two variants of PI, PI with determination of the kernel variance (PID) for white noise and PI with frequency-dependent determination of the variance (PIDF) for colored noise are explained in the next two sections.

3.7.1 Determination of the Kernel Variance for White Noise

To find the mapping function, PI are applied to each sentence in the development set with different values of c and the enhanced recordings are evaluated using the measures of the PEASS toolbox (see Section 4.3.3 for details). As optimization criterion S for the best setting of c, a linear combination of the four scores is used, i.e.,

$$S = 0.5\left(\mathrm{OPS} + \frac{1}{3}\left(\mathrm{TPS} + \mathrm{IPS} + \mathrm{APS}\right)\right). \tag{3.40}$$

Additionally, the IPS score has to be greater than 10 to avoid the situation where S is large due to good TPS and APS scores but no de-noising is achieved. Figure 3.16 shows the scores S as functions of the noise estimate for each sentence of the development set of the *airbone* database. The different curves represent the scores achieved with different values of c during pre-image iterations. The value of c is coded in color. Three conclusions can be drawn from the graphs: First, the noise estimates are scattered along the x-axis so the noise power of utterances in the same SNR condition varies. Second, with increasing noise greater values of c lead to better performance. Third, the overall achievable performance decreases with increasing noise as expected.

The first observation is caused by the fact that at the same SNR the noise power is different for different utterances of the *airbone* database if the active speech level is not considered (see Section 4.1.4 for details). This is the consequence of rather short recordings which lead to a different amount of speech energy from recording to recording and to different noise levels for given SNRs.

Results of the first PI experiments showed that a value of c that is suitable for de-noising is rather related to the noise power than to the overall SNR. This is consistent with the following formulation: Let us consider a part of the spectrum that contains only noise. The degree of de-noising depends on the similarity of the feature vectors. The kernel, which is used to measure the similarity, is scaled by c. If the noise is stronger, the difference between the feature vectors is larger. Therefore, the value of c has to be larger to achieve an equivalent degree of similarity and de-noising.

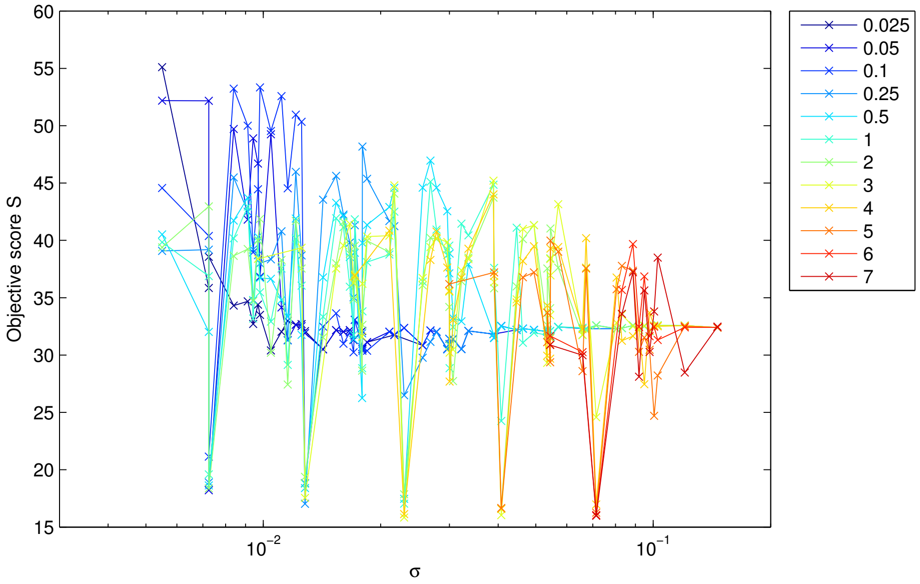


Figure 3.16: Overall performance score S in dependence of the estimated noise level with different values for the kernel variance c. The performance was evaluated on a development set of 12 sentences with an SNR of 0, 5, 10, 15, and 20 dB.

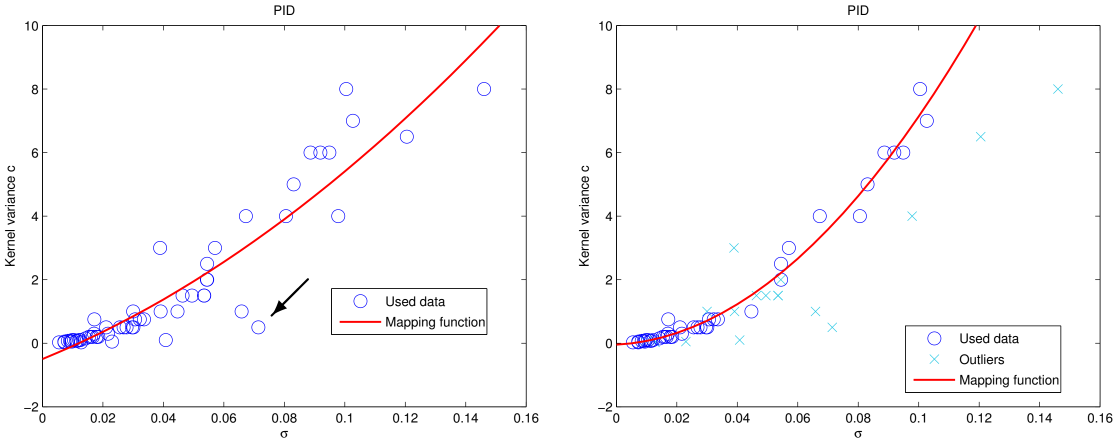
While the noise power gives us the information how to set c for noise attenuation, the SNR is related to the speech signal degradation. If the SNR is high, the speech signal will not or barely be affected by de-noising. If the SNR is low, however, some speech components may not be distinguishable in the noise anymore. These components are attenuated such as noise and the speech signal therefore is distorted.

The mapping function between the noise estimate and the kernel variance c is derived by polynomial curve fitting based on least squares. A polynomial of degree two is used. The fit is computed from the root mean square noise estimate σ. Figure 3.17 (a) shows the fitted function when all data points are used. It can be seen that the fit is not optimal, especially in the regions where the noise power is close to zero.

To improve the fit, outliers are removed. The data points marked by a cross in Figure 3.17 (b) are labeled as outliers since the values of c are not in the appropriate range for the noise estimate. For instance, for the data point marked with the arrow, the SNR is 0 dB and the predicted c is 0.5, which is not reliable as in previous experiments a value around 4 has been identified as a good setting for c at 0 dB. The value of 0.5 rather suitable for 10 dB.

3.7.2 Frequency-Dependent Determination of the Kernel Variance for Colored Noise

For colored noise, a single value for c for all frequencies is not suitable as the noise power is not equally distributed over the frequencies. To approach this problem, we first use a development set with utterances corrupted by white noise to derive



(a) (b)

Figure 3.17: Mapping function for pre-image iterations with automatic determination of the kernel variance c for white noise, (a) before and (b) after removal of outliers.

a mapping function as in the previous section. For the utterances corrupted by colored noise, we estimate the equivalent noise power σ_k^2 at each frequency bin and use it to find a suitable value for c from the mapping function for white noise. This leads to a frequency-dependent estimation of c_k.

The estimate of the noise power for each frequency bin is based on *Parseval's theorem* [45], which states that the mean of the squared magnitude values of the discrete Fourier transform of a signal, $Y[k]$, is equal to the sum of the squared samples in time domain $y[n]$, i.e.,

$$\sum_{n=0}^{K-1} |y[n]|^2 = \frac{1}{K} \sum_{k=0}^{K-1} |Y[k]|^2. \tag{3.41}$$

White noise is equally distributed over all frequencies and if the exact power spectrum could be estimated from one time frame the power spectrum was flat. Therefore, in the ideal case, the power of the time domain signal could be estimated from one Fourier coefficient, i.e.,

$$\sigma^2 = \frac{1}{N}\frac{1}{K} \sum_{k=0}^{K-1} |Y[k]|^2 \overset{\text{ideal case}}{=} \frac{1}{N}|Y[k]|^2 \quad \forall k. \tag{3.42}$$

Based on this relation, we estimate the power spectrum of colored noise and derive the equivalent noise power σ_k^2 in time domain for each Fourier coefficient $Y[k]$.

In particular, 256-point STFTs are computed from 128-sample frames by application of zero-padding. The squared magnitude bins $|Y[k]|^2$ are averaged over the first 15 frames to get a more reliable estimate. Dividing the average by N gives the equivalent noise power σ_k^2 for the k^{th} frequency bin, that is subsequently used to derive a suitable value for c_k from the mapping function. During processing,

Measure	Agreements
Global SNR	6
Segmental SNR (segSNR)	1
Frequency-weighted segmental SNR (fwsegSNR)	0
fwsegSNR variant	1
fwsegSNR MARS[8]	2
Composite measure	0
Perceptual evaluation of speech quality (PESQ)	1
Itakura-Saito distance (IS)	0
Weighted spectral slope distance (WSS)	0
Log-likelihood ratio (LLR)	0

Table 3.2: Agreement counts of cases where objective measures and subjective listening agree on the best parameter setting for 10 dB SNR, evaluated on the development set of the *Noizeus* database (six sentences). Detailed explanations of the evaluation measures can be found in [2].

frequency bins are grouped to frequency bands as explained in Section 3.3. For the frequency bins within one band the values for c_k are averaged and this average is used for pre-image iterations within the band.

The PIDF method was tested on utterances of the *Noizeus* database (see Section 4.1.2 for details) corrupted by car noise. Listening to the enhanced utterances reveals that there is a certain amount of residual noise left when the measure S in (3.40) is used to derive the mapping function. We therefore performed an informal subjective listening test to find the objective evaluation measures that achieve the highest agreement on the perceived degree of de-noising. The test was performed at a single noise condition (10 dB SNR) on the development set of the *Noizeus* database (six speakers). The test person had to find the parameter resulting in good de-noising while preserving good speech quality. The tested measures are listed in Table 3.2. For a detailed description of all measures see [2]. The second column of Table 3.2 shows the counts how often subjective and objective evaluation agreed on the parameter setting leading to the best performance. The table shows that the global SNR achieves the highest agreement with the subjective evaluation. Therefore we use the global SNR instead of S in (3.40) to derive a mapping function for the estimate of a suitable value of c in further experiments. (A comparison of mapping functions is provided in Section 5.3.)

[8] MARS: Multivariate adaptive regression splines

3.8 Pre-Image Iterations for Voice Activity Detection and Musical Noise Suppression[9]

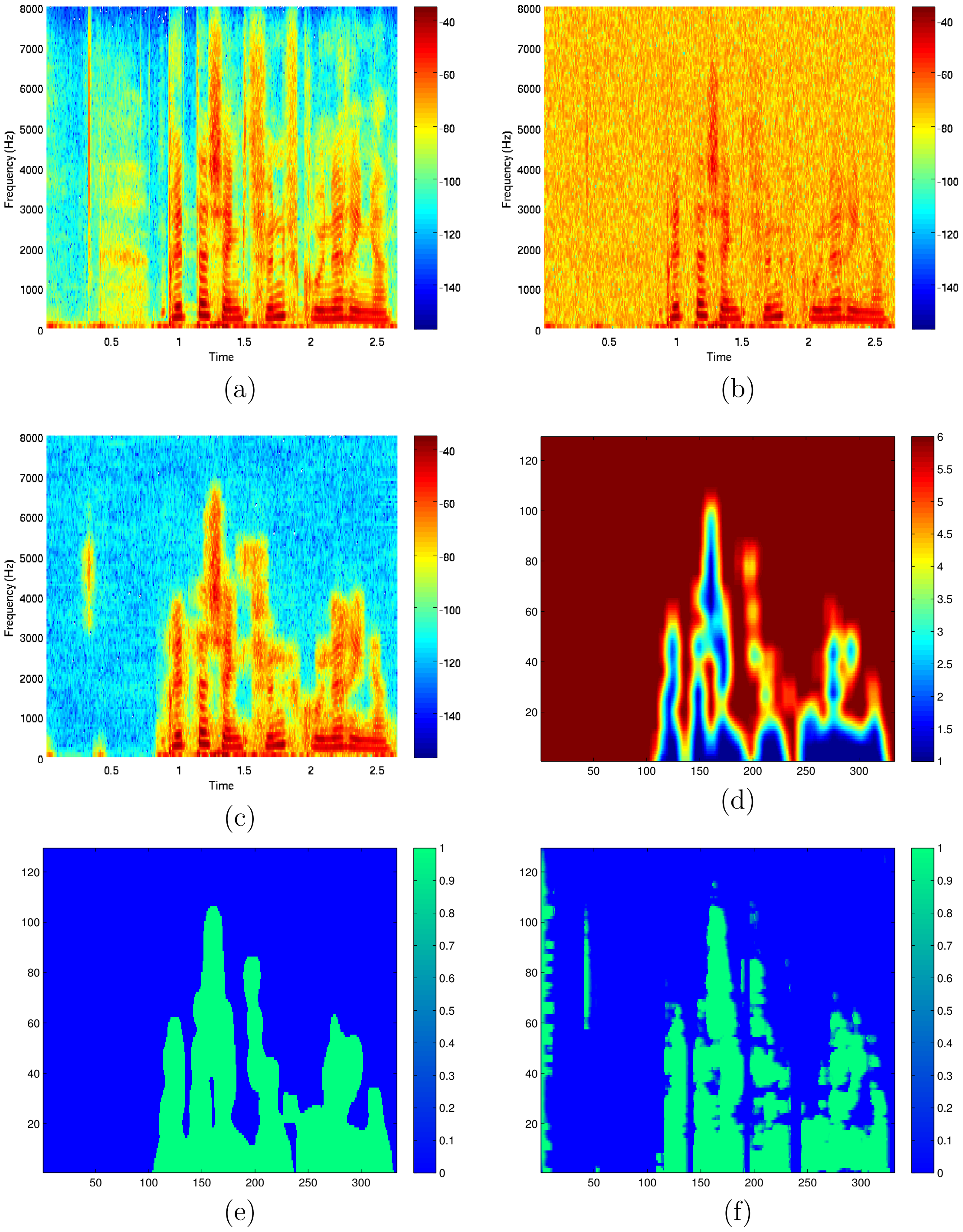
During analysis of pre-image iterations we observed that the convergence behavior of the sample vectors $\mathbf{x}_i$ (or patches, equivalently) reveals information about the content of the underlying signal. To be more precise, the number of iterations until convergence indicates if the sample belongs to a region containing predominantly noise or speech. We use this information for voice activity detection (VAD) and perform musical noise suppression on enhanced speech in a post-processing step.

Figure 3.18 (a) and (b) show the spectrograms of the clean and the noisy signal of a tested utterance. The signal is corrupted by AWGN at 10 dB SNR. Figure 3.18 (c) shows the enhanced signal using PI. Figure 3.18 (d) provides the number of iterations for each bin averaged over the patches the bin belongs to. Figure 3.18 (e) shows the result after thresholding the number of iterations to get a binary decision for voice activity. For comparison, Figure 3.18 (f) shows the result of IMCRA [21], which returns a probability if the voice is active or not for each frequency bin. The figures show that the VAD from pre-image iterations and IMCRA lead to similar results.

Figure 3.19 shows the results of the iteration analysis for the same utterance as in Figure 3.18 but with lower SNRs. The spectrograms of the noisy utterances corrupted by AWGN at 0 and 5 dB SNR are shown in Figure 3.19 (a) and (b), respectively. Figure 3.19 (c) and (d) show the PI VAD and the VAD of IMCRA for 0 dB, Figure 3.19 (e) and (f) show the corresponding figures for 5 dB. In these noise conditions, the VAD of PI and IMCRA are similar as well. However, a comparison of the plots with different SNR makes clear that the kernel variance considerably influences the outcome. Note that the kernel variance is set according to the empirical results on a development set. For 0 dB SNR the VAD seems more robust although the SNR is lower than for 5 dB SNR. For 0 dB the kernel variance is set to 3, which is a rather low value in comparison to the value 4 used in the denoising experiments (see Section 5.1). Obviously, the lower value is more suitable for VAD. These experiments confirm that pre-image iterations can be used for VAD and speech/non-speech separation in the spectrogram in several noise conditions. However, as in the case of de-noising, the value of the kernel variance has to be chosen carefully.

The occurrence of musical noise is a major problem in speech enhancement. Musical noise is caused by inaccuracies of the enhancement algorithm at hand, it originates from a random amplification of frequency bins that change quickly over time. Musical noise is perceived as “twittering” and can severely degrade the perceptual quality of enhanced speech recordings. If it is too prominent, it may even be more disturbing than the interference before enhancement. Figure 3.20 shows the spectrogram of the speech utterance in Figure 3.18 that has been corrupted by additive white Gaussian noise at 10 dB SNR and enhanced by the generalized subspace

[9] This section is partly based on [64] and [65].

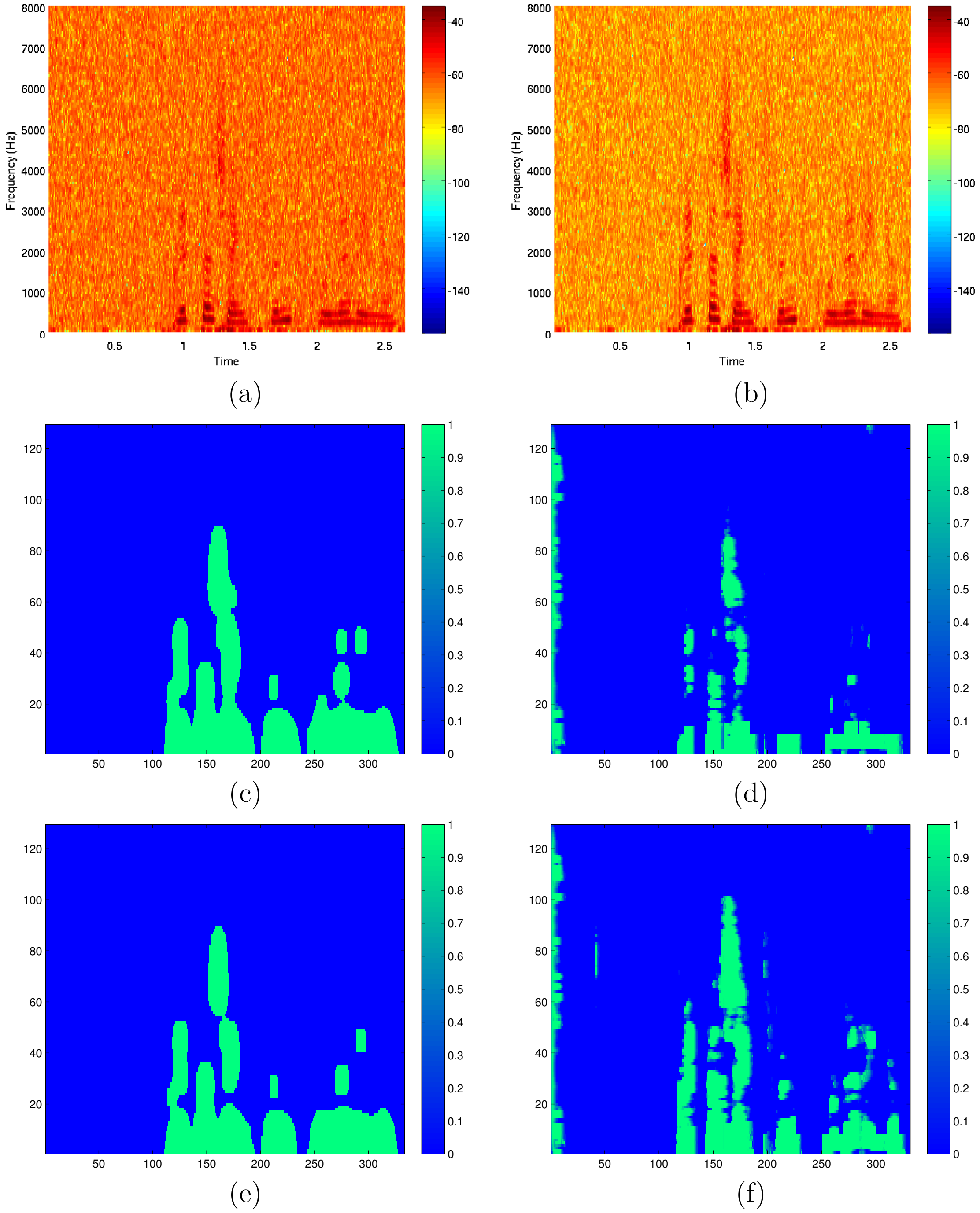


(a) (b)

(c) (d)

(e) (f)

Figure 3.18: Spectrograms of the iteration analysis and voice activity detection for the sample utterance of the *airbone* database. (a) Clean signal. (b) Signal corrupted by AWGN at 10 dB SNR. (c) Spectrogram after enhancement by pre-image iterations. (d) Average number of pre-image iterations until convergence for each bin (with stopping after six iterations). (e) Binary mask after threshold operation. (f) VAD of IMCRA for comparison.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 3.19: Spectrograms and VAD for the sample utterance corrupted by AWGN at 0 and 5 dB SNR. (a) Noisy signal with 0 dB SNR. (b) Noisy signal with 5 dB SNR. (c) VAD of PI and (d) VAD of IMCRA for 0 dB SNR. (e) VAD of PI and (f) VAD of IMCRA for 5 dB SNR. The different performance of the VAD in (c) and (e) is caused by a different value of the kernel variance as explained in the text.

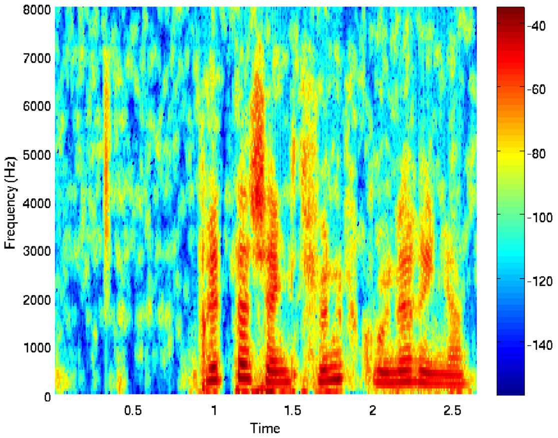


Figure 3.20: The sample utterance shown in Figure 3.18, corrupted by AWGN at 10 dB SNR and enhanced by the generalized subspace method. Musical noise is visible as "blobs" in the spectrogram.

method [40]. The "blobs" in the non-speech region of the spectrogram are perceived as musical noise.

Much research has been carried out on how to combat musical noise, either by modifying the enhancement method at hand or by post-processing. The post-processing method in [66] for spectral subtraction is based on musical noise/speech classification from the spectrum and subsequent processing of the spectral values. In [67], post-filtering with a perceptually inspired filter is applied to the outcome of the used subspace method. The method proposed in [68] can be applied as post-processing for any speech enhancement method, it performs smoothing of weighting gains using a robust detector for speech pauses and low SNR conditions.

We use the VAD derived from the convergence behavior of pre-image iterations to perform musical noise suppression (MNS). Musical noise is most disturbing in non-speech regions. Therefore we use pre-image iterations to discriminate between speech and non-speech regions in the spectro-temporal representation. Then, we apply a mask to attenuate musical noise in non-speech regions. Two application scenarios are proposed: In the first scenario, PI are executed on the noisy signal – in the same manner as for speech enhancement. In the second scenario, PI are executed on the *enhanced* signal. This way, the method can be applied as post-processing step to any speech enhancement algorithm without knowing the original noisy utterance. Both methods are explained in detail in the next sections.

3.8.1 Musical Noise Suppression with PI Applied on the Noisy Signal

In this scenario, we apply a continuous mask to suppress musical noise in non-speech regions of enhanced signals. A continuous mask – in comparison to a binary mask – has the advantage to reduce potential artifacts from inaccuracies of the mask

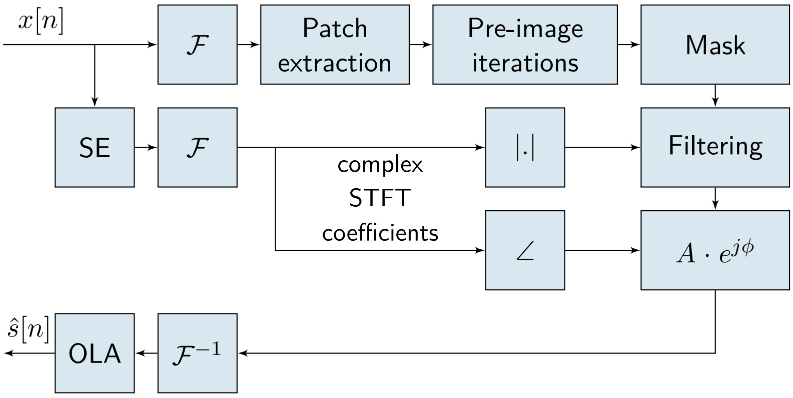


Figure 3.21: Block diagram for musical noise suppression after speech enhancement (SE) with pre-image iterations applied on the noisy signal.

estimation. The block diagram in Figure 3.21 illustrates the method. The mask is computed by application of the sigmoid function

$$m = \frac{1}{1 + \exp(t \cdot a - b)} \tag{3.43}$$

to the number of iterations t for each frequency bin, where a and b are scaling parameters. Figure 3.22 shows the mapping function with the parameters set to $a = 1.2$ and $b = 9$.

To perform musical noise suppression, first the magnitude STFTs of all time frames of the enhanced signal with musical noise $y[n]$ are transformed to the logarithmic domain. Then, the mask of all time-frequency bins is multiplied in element-wise manner with the magnitude STFTs after subtraction of the minimum of all magnitude values. This minimum is added again, the inverse Fourier transform is applied and overlap-add is performed. Figure 3.23 (a) illustrates the resulting mask for the recording plotted in Figure 3.18. Figure 3.23 (b) shows the spectrogram after musical noise suppression performed on the signal enhanced by the generalized subspace method in Figure 3.20. Musical noise is still visible in the spectrogram, however its amplitude is decreased. Listening to the utterance confirms the reduction of musical noise.

3.8.2 Musical Noise Suppression with PI Applied on the Enhanced Signal

To be independent of the noisy signal, we experimented with PI executed on the enhanced signal as illustrated in Figure 3.24. For demonstration, the same test utterance is used as before. Figure 3.25 shows the spectrograms of (a) the noisy signal, (b) the signal after enhancement by the generalized subspace method and (c) the average number of iterations when PI are applied on the enhanced signal. As before, we can discriminate between different regions in the iteration plot, however

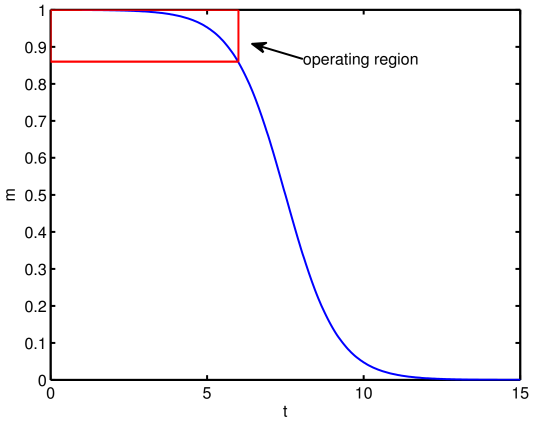


Figure 3.22: Sigmoid mapping function from the number of iterations t to the weight of the mask m. The parameters are set to $a = 1.2$ and $b = 9$. The function is used for maximally 6 iterations (see operating region). Note that the spectrum is transformed to the logarithmic domain and that it is normalized to the minimum. Therefore the resulting suppression is stronger than with simple multiplication by the mask and the suppression factors relatively close to one in the operating region are sufficient.

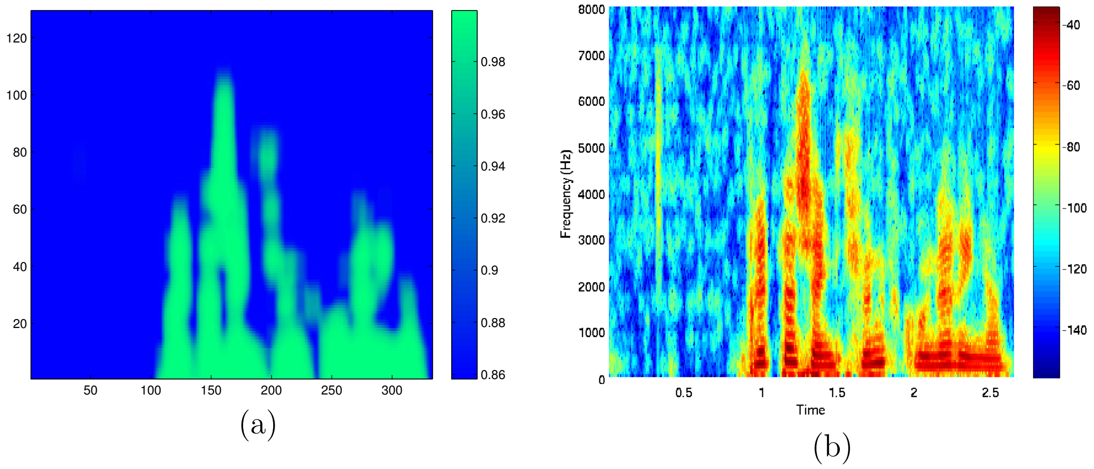


Figure 3.23: (a) Mask computed with the sigmoid function from (3.43) with $a = 1.2$ and $b = 9$ and maximally 6 iterations. (b) Resulting speech utterance with suppressed musical noise after application of the mask.

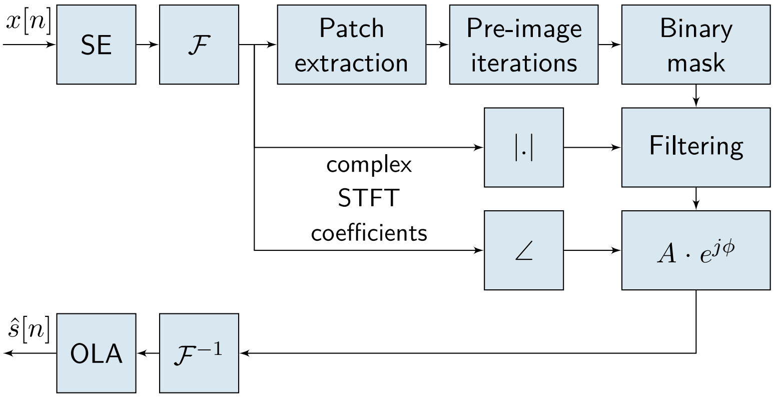


Figure 3.24: Block diagram for musical noise suppression after speech enhancement (SE) based on pre-image iterations applied on the enhanced signal.

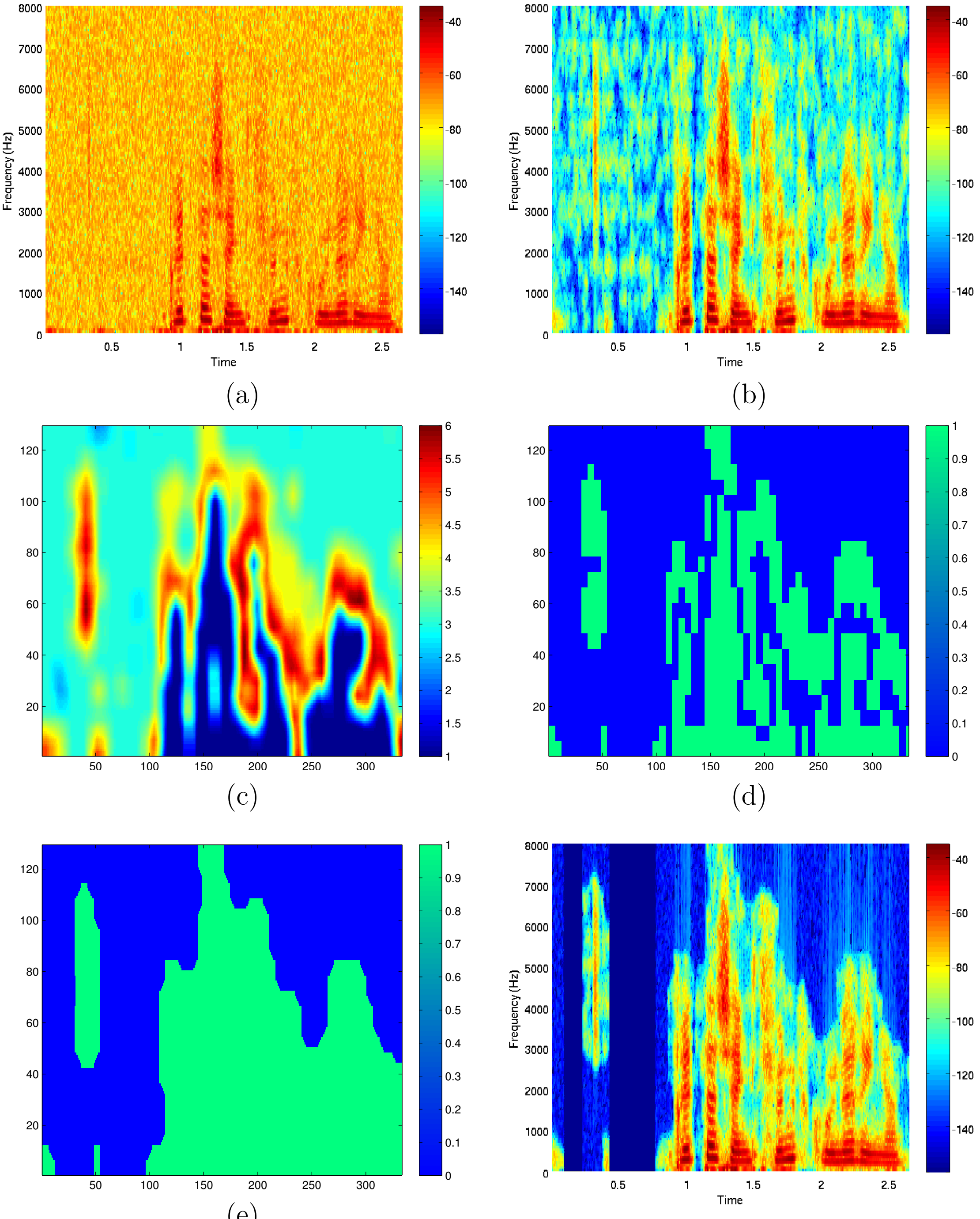
the relation between the number of iterations and the content of the signal is not as clear as in the former case. Empirically, we observed that few iterations correspond mainly to speech regions, an intermediate number of iterations corresponds mostly to noise, and more iterations again correspond to speech regions.

For the discrimination between speech and non-speech regions we set two thresholds, such that the iteration map is segmented as shown in Figure 3.25 (d): Regions in green cover areas corresponding to speech, while regions in blue cover speech and noise areas. The operation for obtaining the mask m for each bin is

$$m = \begin{cases} 1 & \text{if } t < a \text{ or } t > b \\ 0 & \text{otherwise,} \end{cases} \tag{3.44}$$

where $m = 1$ if there is speech, t is the number of iterations for a specific bin in the map and a and b are the two threshold values. The values for the thresholds are derived experimentally (see Chapter 5, Section 5.4).

To distinguish between noise and speech, the parts of the blue region within speech areas have to be removed. This is realized with techniques from image processing, namely morphological filtering such as dilation and erosion [69]. The consecutive execution of these operations results in the so-called closing operation that closes the holes in Figure 3.25 (d). As structural element a disk of radius 10 is used. Figure 3.25 (e) shows the resulting contiguous binary mask, which is subsequently applied to filter the magnitude of the STFT of the signal in the same way as explained in Section 3.8.1. Figure 3.25 (f) represents the spectrogram after musical noise suppression. A comparison of the resulting spectrograms after musical noise suppression by the two methods shows, that the application of the binary mask leads to stronger suppression with the given parametrization.



(a) (b) (c) (d) (e) (f)

Figure 3.25: Spectrograms of the iteration analysis and suppression mask of the sample utterance of the *airbone* database. (a) Signal corrupted by AWGN at 10 dB SNR. (b) Signal enhanced by the generalized subspace method. (c) Average number of iterations for each frequency bin computed from the enhanced signal (with stopping after six iterations). (d) Binary mask after the threshold operation (3.44). (e) Smoothed mask after the closing operation. (f) Spectrogram of the signal after musical noise suppression.

Chapter 4

Experimental Setup and Evaluation

To test the proposed algorithms, enhancement experiments were performed on two databases: the *airbone* database and the *Noizeus* database. The speech enhancement methods are evaluated using objective quality measures, by listening, by visual inspection of the spectrograms, and by automatic speech recognition (ASR). For the speech recognition experiments, a recognizer was trained on the BAS PD1 database [70]. As a benchmark, the proposed methods are compared to the generalized subspace method, to spectral subtraction with oversubtraction and spectral flooring, and to the MMSE log-spectral amplitude estimator.

In this chapter, we first give an overview on the databases, then we discuss the evaluation including a summary of applied objective quality measures and a description of the speech recognizer. Finally, we provide a summary about the methods used for reference.

4.1 Databases

Three databases were used: The *airbone* database and the BAS PD1 database contain recordings in German and the *Noizeus* database contains recordings in English.

4.1.1 Airbone Database

The *airbone* database consists of utterances recorded with a headset supplied with a bone conduction microphone in addition to the standard air conduction microphone, hence the name *airbone* database. The bone microphone is placed at the temporal bone behind the ear and captures sound waves propagating through the cranial bones. The bone channel is robust to environmental noise and can therefore be used to improve the performance of speech processing applications in noisy environments. The air and the bone channel are directly processed as stereo recording by using a specially developed recording device. A prototype of this device was built at the lab [71] and is shown in Figure 4.1.

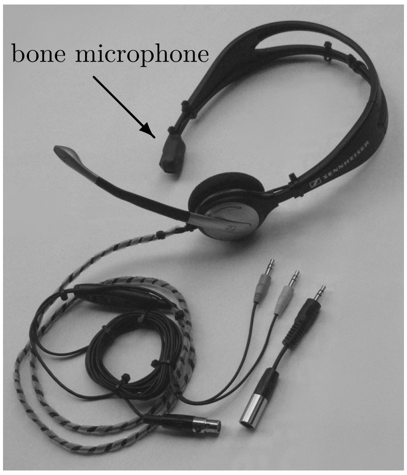


Figure 4.1: Headset with integrated bone conduction microphone [71].

The database consists of recordings of six individuals, three female and three male, speaking the Austrian variety of German. Each speaker read a list of 20 sentences that were randomly generated from a word list of 50 words. The sentences have the grammatical structure *subject verb numeral adjective object*. The phoneme distribution of the basic list is consistent with the phoneme distribution of the German language.

The recordings were performed with 16 kHz sampling frequency. To avoid clipping they were normalized to -2 dBFS[1]. The recordings of two speakers (speaker 1 and 3) are corrupted by a hum at 50 Hz, presumably caused by the recording device. As the disturbing signal is well below the frequency range of the speakers, who are both female, it was removed by filtering with a high-pass filter.

AWGN and car noise were added at 0, 5, 10, and 15 dB SNR. For a subset of the experiments noise was added depending on the *active speech level* (ASL) (see Section 4.1.4) while for the other experiments the SNR was computed using the entire recordings. Note that the experiments reported in this work make only use of the air channel.

4.1.2 Noizeus Database

Noizeus is a speech corpus developed at UT Dallas to enable comparison of speech enhancement algorithms among different research groups [2, 72]. The database contains recording of 30 IEEE sentences [73] spoken by 6 speakers, three female and three male, each producing five sentences. The utterances were corrupted by eight different real-world noises, which were taken from the AURORA database [74]. The noise types comprise suburban train, babble, car, exhibition hall, restaurant, street, airport, and train-station noise. The IEEE sentences were selected because they are phonetically balanced and have relatively low word-context predictability.

[1] dB full scale

The sentences were recorded in a sound-proof booth and with a sampling frequency of 25 kHz. The recordings were down-sampled to 8 kHz and then filtered by the modified intermediate reference system (MIRS) filters used for the PESQ measure (see Section 4.3.1) to simulate the frequency characteristics of a telephone handset. The noise was filtered by the MIRS independently of the speech signal. To determine the SNR, the ASL of the filtered clean speech signal was computed using method B of the ITU-T recommendation P.56 [75]. Noise was added at 0, 5, 10, and 15 dB SNR. To add the noise, a noise segment of the same length as the speech utterance was randomly cut out of the noise recording, rescaled according to the ASL and the desired SNR, and added to the filtered speech signal.

We used the data contaminated by car noise and additionally corrupted clean recordings by AWGN for the experiments with the PIDF method.

4.1.3 BAS PhonDat 1 Database

The *BAS PhonDat 1* (BAS PD1) database belongs to the *Bavarian Archive for Speech Signals Corpora* [70]. The database was created to have access to different regional variants of German, for both documentation of phonological forms and improvement of speech processing systems, e.g., for ASR. Therefore, the recording was performed at four sites in Germany (Kiel, Bonn, Bochum, Munich). The BAS PD1 corpus contains read speech uttered by 201 different speakers of German. In total 21587 utterances were recorded with a sampling rate of 48 kHz. The data was downsampled to 16 kHz. The entire database is phonologically segmented by automatic segmentation.

4.1.4 SNR Computation

For adding noise to the speech signals of the *airbone* database, we employed two different approaches of SNR computation. The first is based on the power of the entire clean signal and the second is based on the ASL as it is done for the *Noizeus* database.

The global SNR is computed by

$$\mathrm{SNR} = 10\log_{10}\frac{\sum_{n=0}^{N-1}|s[n]|^2}{\sum_{n=0}^{N-1}|d[n]|^2}, \tag{4.1}$$

where $s[n]$ is the speech signal, $d[n]$ is the noise signal and N is the length of the signals [76]. For computation based on the ASL, only the frames with active voice are taken into consideration. The power estimate for the signal is higher when only speech frames are used instead of the entire signal including silent regions with relatively low power. Hence, the measured global SNR is lower when the SNR computation is based on the ASL.

To get an estimate of the difference in SNR, we computed the power with and

	d_{SNR} **[dB]**			
Database	**Mean**	**Std. dev.**	**Minimum**	**Maximum**
airbone	1.7838	0.3613	0.7835	2.7083
Noizeus	0.5547	0.2007	0.2983	1.1246

Table 4.1: SNR difference d_{SNR} between SNR computation with and without ASL detection.

without ASL detection and derived the difference in SNR as

$$d_{\mathrm{SNR}} = 10 \log_{10} \left(\frac{\sigma^2_{\mathrm{ASL}}}{\sigma^2} \right). \tag{4.2}$$

Table 4.1 shows the d_{SNR} averaged over all utterances of the *airbone* and the *Noizeus* database. For the *airbone* database the average difference is larger. This is reflected by the speech activity factor, which gives a percentage for the amount of active speech. For the *airbone* database the average activity factor is 0.67 while for the *Noizeus* database it is 0.88. This implies that the *airbone* database contains more silent periods, i.e., frames where speech is not active. Consequently, the difference between the two SNRs is larger than for the *Noizeus* database.

4.2 Evaluation of Speech Enhancement Methods

The objective of speech enhancement can be manifold. One objective is to improve perceptual aspects such as the perceived speech quality or the intelligibility. A different objective is to enhance speech signals not for humans but for machines, such as for ASR. For all of these tasks, a speech enhancement method with good performance for one task may not be performing as well on a different task. For instance, speech can be highly intelligible while of poor quality or, on the other hand, of good quality but not fully intelligible [2]. Speech quality and intelligibility are not equivalent and the relationship between the two is not yet fully understood [77]. For ASR, enhancement has an even different objective, namely the increase of recognition rates by improving the features for the recognizer.

Accordingly, evaluation of speech enhancement methods can be performed with the focus on different aspects. Perceptual evaluation covers speech quality and intelligibility. Speech quality is highly subjective and therefore difficult to evaluate. This is caused by the different internal standards of individual listeners for what is considered to be “good” or “bad” quality. Quality evaluation can be done either by subjective listening tests or by objective quality measures. For subjective evaluation, listeners are asked to rate the quality of original and processed speech along a given scale. Listening tests have to be designed with caution such that valid conclusions can be derived [78]. As carefully designed listening tests are time consuming and therefore expensive, there is a great interest in objective quality measures, which are easier to apply and faster. Objective evaluation is done by comparing the processed speech signal with the original (clean) signal by mathematical means. Objective quality measures are derived from the numerical "distance" between processed and original signals. An objective quality measure can only be valid if it correlates well with the outcome of subjective listening tests. Therefore, research is ongoing to investigate the correlation of objective quality measures and listening tests [79, 63].

While quality measures "how" a speech utterance is produced, intelligibility relates to "what" a speaker said. In contrast to quality, speech intelligibility is not subjective and can be easily assessed by asking a group of listeners to identify words in given speech material. The intelligibility is then measured by counting the number of correctly identified words or phonemes.

Besides the discussed enhancement with focus on perceptual aspects, speech can be optimized for automatic processing by machines. If an automatic speech recognizer is trained on clean data, its performance in general suffers if noise is present in the tested signal. One possibility for improvement is to enhance the speech signal before recognition. Note that high ASR rates do not necessarily indicate good perceptual quality.

4.3 Objective Quality Measures

As listening tests are time consuming and expensive and often require access to trained listeners [2], much afford has been made to develop objective quality measures that provide high correlations with the scores of subjective listening tests [76]. Most of these measures were originally designed to measure distortions introduced by speech codecs and/or communication channels. These distortions are, however, different from the distortions introduced by speech enhancement algorithms. In the case of speech enhancement, two types of distortions can be observed: the distortion of the speech signal component and the distortion by the background noise. This is caused by the suppression of the background noise that also affects and possibly degrades the speech signal [79]. For instance, speech components may erroneously be attenuated.

In [79], Hu and Loizou present experiments that analyze the correlation between results of subjective listening tests and objective quality measures. For this purpose the speech utterances were enhanced by 13 different enhancement algorithms and subsequently tested by subjective listening tests. The listening tests were performed according to the ITU-T P.825 standard (see Section 4.4.3) and correlations were computed for signal distortion, background distortion and overall quality. Among the tested quality measures, the *perceptual evaluation of speech quality* (PESQ) measure showed the highest correlation with the overall quality and the signal distortion from listening tests. The *log-likelihood ratio* (LLR) and the *frequency-weighted segmental SNR* (fwsegSNR) achieved an almost similar performance. As they are easier to compute than the PESQ measure, they represent simple alternatives.

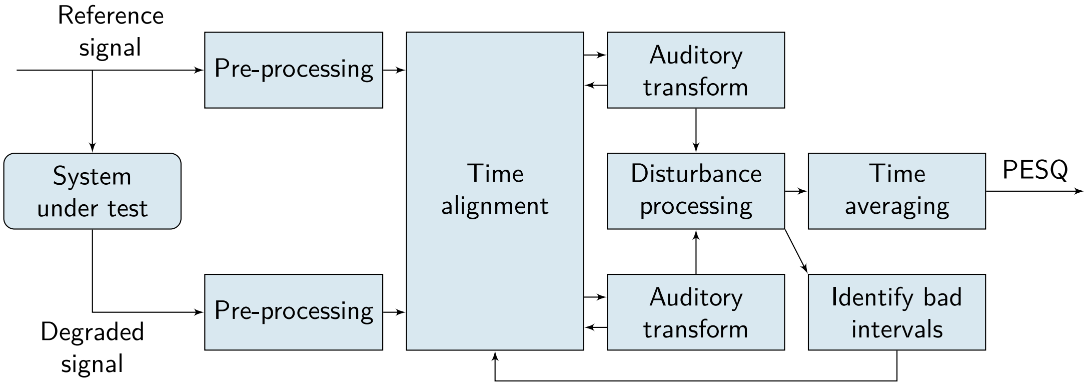


Figure 4.2: Block diagram of the PESQ measure computation [2].

A comprehensive overview on objective quality measures for speech enhancement is given in [2]. In the next sections, we shortly summarize the PESQ, the fwsegSNR, and the perceptual quality measures of the PEASS toolbox for source separation as we use them for evaluation.

4.3.1 Perceptual Evaluation of Speech Quality Measure

The PESQ measure was developed for evaluation of distortions in real telecommunication networks. These include packet loss and signal delays in VoIP or linear filtering and coding distortions [80, 2]. In the 1990s, several objective measures addressing these problems were proposed [81, 82, 83]. A competition was held in 2000 to find a measure that reliably performs in many codec and network conditions. The competition was jointly won by the perceptual analysis measurement system (PAMS) [81] and by PSQM99, an unpublished version of the perceptual speech quality measure (PSQM), which was standardized in ITU-T P.861 [84]. Parts of these two measures were combined to a new measure called *perceptual evaluation of speech quality* (PESQ) measure [85]. PESQ was standardized as ITU-T P.862 [86]. In the ITU-T evaluation, a high correlation with subjective listening tests was found for both known and unknown test data (average Pearson correlation coefficient of 0.935 [86]).

Figure 4.2 shows the structure of the PESQ measure computation. First, the reference (clean) signal and the degraded (processed) signal are equalized to a standard listening level. Then, the signals are filtered by the MIRS filter in the pre-processing stage. The signals are aligned in time to compensate for delay errors and then transformed by an auditory transform that returns loudness spectra. Then, the absolute difference between the degraded and the reference spectrum is computed and further used as an error measure in the next processing stage. Note that the computation of the difference is not symmetric as in other measures, where the difference is squared. Positive and negative differences are treated differently. This is based on the observation that positive and negative differences influence the perception differently. While a positive difference indicates the addition of a component, such as noise, a negative difference indicates that a component has been attenuated. In telecommunication applications, attenuation is considered to be less objectionable than additive components, i.e., the omitted components may not be perceivable due to masking effects. The differences between loudness spectra, called disturbances, are averaged over time and frequency. The final mean opinion score (MOS, see Section 4.4.1) is derived by linear combination of the symmetric disturbance value and the asymmetric disturbance value. The PESQ measure covers the range from -0.5 to 4.5, however, for most cases the score will be between 1 and 4.5 such as the MOS.

Although designed for speech codecs and assessment of transmission errors, PESQ was reported to show high correlation with the outcome of subjective listening tests on speech enhancement algorithms in the study of Hu and Loizou [79]. They therefore recommended it for evaluation of speech enhancement algorithms.

4.3.2 Frequency-Weighted Segmental SNR

Besides the global SNR in (4.1), which is evaluated on the entire signal, the SNR can be computed for frames that are subsequently averaged leading to the *segmental SNR*. In addition to the evaluation in time domain, the segmental SNR can also be evaluated in frequency domain. Hu and Loizou showed in [79] that the segmental SNR in time domain yields a very poor correlation coefficient with the overall quality. The frequency-weighted segmental SNR (fwsegSNR) achieves a higher correlation – similar to the PESQ measure – and is therefore more suitable for evaluation.

The fwsegSNR is defined as

$$\text{fwsegSNR} = \frac{10}{N} \sum_{n=0}^{N-1} \frac{\sum_{k=1}^{K} B_k \log_{10} \left[\frac{F^2(n,k)}{(F(n,k)-\hat{F}(n,k))^2} \right]}{\sum_{k=1}^{K} B_k} \tag{4.3}$$

where B_k is the weight of the k^{th} frequency band, K is the number of bands, N is the total number of frames in the signal, $F(n,k)$ is the filter-bank amplitude of the clean signal in the k^{th} frequency band at the n^{th} frame, and $\hat{F}(n,k)$ is the filter bank amplitude of the enhanced signal in the same band. The fwsegSNR is superior to the time-domain segmental SNR by providing additional flexibility to choose the weights for different bands of the spectrum. For instance, perceptually motivated frequency spacing can be applied such as critical band spacing or the weighting can be chosen in order to achieve maximal correlation with the results of subjective listening tests [2].

For the evaluation of our experiments, we used octave bands with the weights for the speech transmission index reported in [87].

4.3.3 Objective Quality Measures Proposed for Source Separation

Audio source separation is a technique related to speech enhancement. In speech enhancement one target speaker signal is extracted from a noisy speech recording, while in audio source separation one or more sources are separated from a mixture of sources. Depending on the “noise”, which – broadly defined – can be any interfering signal, both tasks have a similar goal. The main difference is that in speech enhancement only one speaker is retrieved while in source separation the objective is in general to retrieve more signals separately.

Similar as in speech enhancement, source separation requires evaluation by subjective listening tests or by objective quality measures proven to correlate well with such tests. In [63], a set of new objective quality measures for audio source separation was proposed. These measures are based on the estimation of distortion components and the use of the *perception model quality assessment* (PEMO-Q) auditory model [88] to derive salience features of the overall distortion and each distortion component. These features are non-linearly combined by a neural network to optimally match the scores of subjective listening tests. This results in four scores testing different aspects of the processed signal: the global quality (OPS - overall perceptual score), the preservation of the target signal (TPS - target perceptual score), the suppression of other signals (IPS - interference perceptual score), and the absence of additional artificial noise (APS - artifact perceptual score).

The measures were developed and evaluated using the results of subjective listening tests. The performance evaluation showed a high correlation to subjective scores and an improvement compared to the state-of-the-art evaluation measures such as signal-to-distortion ratio (SDR), source image-to-spatial distortion ratio (ISR), signal-to-interference ratio (SIR) and signal-to-artifacts ratio (SAR) [89]. The set of measures and source code for evaluation is publicly available as *Perceptual Evaluation Methods for Audio Source Separation* (PEASS) toolbox [63].

Although originally proposed for source separation, we use these measures to evaluate the performance of speech enhancement algorithms as the two domains are closely related. The interference signals used in [63] are speech and music, so the application to our scenario with white and car noise as interference is probably not optimal. However, listening to the decomposed audio signals containing target, interference and artifact signal provided by the PEASS toolbox confirmed that the decomposition works well. One open question is if the non-linear mapping from the salience features to the scores is optimal for speech enhancement, as there may be a mismatch between the data used to derive the mapping and the data of the speech enhancement experiments. Despite this issue, the usage of the PEASS measures is interesting, because they provide better insight to the effects of different algorithms due to the computation of the four scores. This is not possible with evaluation measures that only return one overall quality score such as PESQ.

4.4 Subjective Listening Tests

To gain full understanding about the perceived speech quality after processing by speech enhancement methods, a perceptual listening test is necessary. We, however refrained from doing so. In [78], the authors suggest to carefully investigate if it

Rating	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible, but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

Table 4.2: Mean opinion score for sound quality evaluation [90].

is efficient to conduct a listening test to answer a given research question. Among other considerations, they suggest that no perceptual listening test is necessary if the magnitude of perceptual differences is either very large or very small. If the magnitude is expected to be large, the outcome of the test might be obvious so no verification is needed. On the other hand, if the differences are very small a large number of subjects or repetitions will be necessary to gain information that obeys a certain statistical confidence level. The results of the objective quality measures we use for evaluation indicate that the perceptual difference between the investigated methods is rather small. Applying the arguments in [78], a perceptual evaluation would need too much effort to achieve results with a certain statistical confidence.

Though we did not realize a formal listening test, for completeness we give a short overview on experimental setups that are commonly employed to evaluate speech enhancement algorithms. For audio evaluation in general, the authors of [78] differentiate between *perceptual* and *affective* evaluation. In *perceptual* evaluation, certain attributes of the tested stimuli are rated. These attributes have to be well defined and chosen by a proper procedure in advance of the execution of the experiment. In *affective* evaluation the preference for stimuli is evaluated without explicitly looking at certain attributes of the stimulus. The listener is asked to rate "overall" impression about the stimuli. The listener is assumed to be in an integrative state of mind, where several factors are combined to one overall impression of the stimulus. Influencing factors contain of course the individual perceived attributes, but also the mood of the listener, the context, the previous experience, and the expectation.

Note that in contrast to [78] we refer to all methods involving subjective tests as perceptual evaluation. In the next sections, we will describe tests that are suitable to assess the quality of enhanced speech such as the mean opinion score, preference tests, and tests that are designed to assess different attributes of the enhanced signal such as the diagnostic acceptability measure.

4.4.1 Mean Opinion Score

The Mean opinion score (MOS) is widely used to rate the speech quality [2]. The listeners are required to judge the quality on an absolute scale consisting of five points from 5 meaning "excellent" to 1 denoting "bad" quality (see Table 4.2). The

Rating	Description
5	Very natural, no degradation
4	Fairly natural, little degradation
3	Somewhat natural, somewhat degraded
2	Fairly unnatural, fairly degraded
1	Very unnatural, very degraded

Table 4.3: Signal rating scale of the ITU-TP.835 standard [95].

scores of all listeners are averaged and the resulting MOS is taken as measure for the quality. The MOS is recommended by the IEEE Subcommittee on Subjective Measurements [73] and by the ITU [91, 92, 90].

Generally, expert listeners are preferred to non-expert listeners. A listening test using the MOS is conducted in two phases: First the listeners undergo a training phase, then the actual test is executed. The training phase is required to familiarize the listeners with the test procedure, material and environment. In the test phase, the test signal is presented to the listeners who are asked to rate it according to the five categories listed in table 4.2. More detailed guidelines on the test procedure are provided in the ITU-R BS.1284-1 standard [90].

4.4.2 Diagnostic Acceptability Measure

Besides the overall quality of an enhanced speech signal, it is often useful to ask listeners for certain attributes of the signal. In [93], a multidimensional approach based on the diagnostic acceptability measure (DAM) was proposed. This test is motivated by the work of McDermott [94]. The DAM test contains evaluation on three scales, the parametric, metametric and isometric scale. The metametric and the isometric scale assess the quantities “intelligibility”, “pleasantness”, and “acceptability”. The parametric scale measures signal and background distortions. The scales result in 16 measurements in total, containing several attributes that are evaluated on the speech signal and on the background. For instance, the listener is asked to rate on a scale from 0 to 100 how muffled the speech signal sounds while ignoring the background distortion. The listener is also asked to quantify on a scale from 0 to 100 if there is hissing, buzzing, chirping, or rumbling perceivable in the background. The DAM test requires trained listeners and therefore is time consuming.

4.4.3 The ITU-T P.835 Standard

The MOS and DAM tests were originally designed for the evaluation of speech coders. Speech enhancement algorithms, however, have different effects on a speech signal than speech coders [2]. Typically, besides the suppression of background noise

Rating	Description
5	Not noticeable
4	Somewhat noticeable
3	Noticeable but not intrusive
2	Fairly conspicuous, somewhat intrusive
1	Very conspicuous, very intrusive

Table 4.4: Background rating scale of the ITU-T P.835 standard [95].

they also also influence the speech signal. This makes the subjective evaluation difficult as it is not clear whether the rating of a listener is based on the signal distortion, the background suppression, or both. The lack of knowledge about the listeners motivation for the rating introduces an uncertainty of the measures and reduces the reliability of the results. The ITU-T P.835 standard was designed to overcome this issue. The listener is asked to rate only the speech signal, only the background noise, and the overall effect of speech and noise quality alternately. To be precise, each trial contains a three-sentence sample of speech, where each sample is followed by an appropriate silent interval for rating. In the first half of the experiment, the listener rates “signal – background – overall quality”, in the second half ”background – signal – overall”. This procedure is applied to prevent influence by the scale order. The rating scales for the signal quality and the background intrusiveness are given in Table 4.3 and 4.4, respectively. For the third rate in each trial, the overall quality scale used in MOS tests is applied (Table 4.2).

4.4.4 Preference Tests

Preference tests are one example for affective evaluation according to [78]. Pairs of samples are presented to listeners and they are asked to express their preference for one sample. The pair of samples can either contain one modified signal and one reference signal, or signals modified by two different systems.

The probably simplest form of preference test is the forced-choice paired comparison test. Test samples of two systems A and B are presented to listeners and they have to choose which signal they prefer. As result, the percentage of preference votes for system A is given [2]. Although a paired comparison test tells us whether system A or B is preferred, it does not provide information about the degree of preference [2]. The comparison category rating (CCR) test assesses the preference of one system over the other on a seven-point scale including ratings for positive and negative preference [92]. These ratings are listed in Table 4.5.

In contrast to the above preference tests, the method described in [96] requires reference signals. The method is based on the comparison of the test signal with five differently distorted speech signals as reference. Pairs including all combinations of reference signals and all combinations of the test signal with the reference signals

Rating	Quality of second stimulus compared to first
3	Much better
2	Better
1	Slightly better
0	About the same
-1	Slightly worse
-2	Worse
-3	Much worse

Table 4.5: Comparison category rating scale [92].

are presented to the listeners. The listeners are asked to express their preference. The ratings obtained by comparison of the test signal with the reference signals are plotted against the ratings obtaind by comparison of the reference signals with all other reference signals. This way, the reference signal that is of equal preference as the test signal can be found [96]. This method is recommended for speech quality measurements by the IEEE Subcommitee on Subjective Measurements [73].

In the context of paired comparison tests, Bech and Zarachov mention the "law of comparative judgments" [78]. It is based on the assumption that the comparison leads to a distribution of responses for each stimulus. From these distributions, a scale with scores for each stimulus can be derived. To be valid, however, one basic assumption has to be fulfilled: "the unidimensionality of the attribute continuum that is assumed to exist in the mind of the subject" [78]. This means, that it is possible to represent the perceived differences of the stimuli on a single scale. This further implies, that all derived differences between stimuli can be represented on a single scale as well. So, if a stimulus A is preferred to B and B is preferred to C, then A should be preferred over C.

In speech enhancement, we encounter two types of signal degradations: The background noise and the signal distortion due to processing. Now, assume an experiment where the following signals are presented to a listener: (i) a signal without noise suppression, (ii) a signal processed by weak noise suppression resulting in some de-noising with no effects on the speech components, (iii) a signal processed by strong de-noising with distortion of speech components. It is questionable if the speech samples are rated such that the scores lie on a scale that is proportional to the degree of processing. For instance, the speech sample with strong processing might obtain lower preference than the unprocessed sample due to disturbing artifacts, while the weakly processed sample might by preferred over the unprocessed sample as noise is reduced. Therefore, the assumption of unidimensionality is possibly not fulfilled and a preference test probably is not the best method to assess the quality of speech enhancement methods.

4.5 Automatic speech recognition[2]

In ASR, recognition rates of recognizers trained on clean data generally drop when data is corrupted by noise due to the mismatch between the clean training condition and the noisy testing condition. Speech enhancement has the potential to improve the recognition results if noise is removed properly. However, enhancement methods that result in good perceptual quality may not be optimal for ASR. In reverse, it is not generally valid to expect good quality from good recognition rates on noise-contaminated data. We performed some experiments reported in Section 5.5 to test whether our speech enhancement methods are suitable to improve speech recognition rates.

To test the enhanced utterances of the *airbone* database, we use a speech recognizer trained on clean data of the BAS database (see Section 4.1.3). For training, 4999 sentences of the BAS database, spoken by 50 speakers were used. This amounts to around 100 sentences per speaker. The training set contains 1504 different words and the test set of the the *airbone* database 50, which do not necessarily coincide with each other.

The automatic speech recognizer is based on the *Hidden Markov Toolkit* (HTK) [97]. The front-end (FE) and the back-end (BE) are both derived from the standard recognizer of the Aurora-4 database [98]. The FE computes Mel frequency cepstral coefficients (MFCCs) by using a sampling frequency of 16kHz, a frame shift of 10 ms, a window length of 32 ms, 1024 frequency bins, 26 Mel channels, and 13 cepstral coefficients. Cepstral mean normalization is employed on the MFCCs. Furthermore, delta and delta-delta features are computed with a window length of 5 (half length 2). This finally leads to a feature vector of 39 components.

For training, the BE uses a dictionary based on 34 SAMPA-monophones. The transcriptions in this dictionary are derived from more detailed transcriptions based on 44 SAMPA-monophones by clustering of monophones that are less common in the corpus. For each triphone, a hidden Markov model (HMM) is trained, which consists of 6 states and Gaussian mixture models of 8 components per state. To reduce the complexity and to overcome the lack of training data for some triphones, a tree-based clustering based on monophone-classification is applied. With tree-based clustering also triphone models that have not been observed in the training data can be created. The grammar used for training is probabilistically modeled. In contrast to that, a rule-based grammar is applied for testing as the utterances of the *airbone* database obey very strict grammar rules.

4.6 Reference Methods

For evaluation, we compared our methods to spectral subtraction with oversubtraction and spectral flooring [10], the generalized subspace method [40], and to the

[2] The evaluation of our speech enhancement methods with ASR is joint work with Juan A. Morales Cordovilla.

MMSE log-spectral short-time amplitude estimator [99]. Spectral subtraction is used because it is simple in implementation. The generalized subspace method was tested because it provides the basis of applying kernel PCA for speech enhancement. The MMSE log-STSA estimator was taken to include a statistical model-based algorithm in the comparison. The implementations of all three algorithms are provided in [2]. As far as not noted differently, we use the default parametrization.

The noise is estimated from the beginning of each recording, where speech is assumed to be absent. Each method includes a simple VAD for noise updates in speech pauses. We further tested more sophisticated noise estimation algorithms in combination with the MMSE log-STSA estimator, namely, the IMCRA method [21] and the minimum statistics method [19]. These methods, however, did not improve the quality (perceived subjectively and measured objectively). The tested noise types are stationary, so the noise tracking methods probably have no advantage over estimating the noise at the beginning with updates in speech pauses.

4.6.1 Spectral Subtraction with Oversubtraction and Spectral Flooring

Spectral subtractive methods are based on the assumption that noise is additive. Hence, the noisy speech signal can be enhanced by subtracting a noise estimate from the noisy speech in the spectral domain. If the noise estimate is not exact, residual noise is left that mostly appears as tonal components changing from frame to frame, as the gain function applied on the noisy speech signal is time-varying. This is called musical noise. Berouti et al. [10] proposed a spectral subtraction method less prone to musical noise. The reduction of musical noise is addressed by two measures: First, an overestimate of the noise is subtracted, this leads to better suppression of noise components. Second, the noise is prevented to drop below a certain noise floor. This way, gaps between remaining residual noise peaks are filled and the residual noise has rather the characteristic of broad-band noise than musical noise.

The algorithm suggested by Berouti et al. computes the enhanced power spectrum $|\hat{X}(\omega)|^2$ as

$$|\hat{X}(\omega)|^2 = \begin{cases} |Y(\omega)|^2 - \alpha|\hat{D}(\omega)|^2 & \text{if} \quad |Y(\omega)|^2 > (\alpha+\beta)|\hat{D}(\omega)|^2 \\ |\hat{D}(\omega)|^2 & \text{else,} \end{cases} \tag{4.4}$$

where $|Y(\omega)|^2$ is the power spectrum of the noisy speech signal, $|\hat{D}(\omega)|^2$ is the estimate of the noise power spectrum, α is the oversubtraction factor ($\alpha \geq 1$) and β is the spectral floor parameter ($0 < \beta \ll 1$). For inverse transformation to time-domain, the phase of the noisy speech signal is used [2].

The parameter α determines the trade-off between speech distortion and residual noise. Hence, it is varied from frame to frame depending on the estimated SNR such that

$$\alpha = \alpha_0 - \mathrm{SNR}/s \quad \text{for} - 5\text{ dB} < \mathrm{SNR} < 20\text{ dB}, \tag{4.5}$$

where α_0 is the value of α at SNR $= 0$ dB and $1/s$ is defined as the slope of the linear function to determine α. In [10], it is recommended to set α to 1 for SNR > 20 dB, and to keep α fixed for SNR < -5 dB.

In our experiments, the parameter α_0 is set to 4 and $1/s$ is set to 20/3, hence $\alpha = 4.75$ for SNR ≤ -5 dB. The spectral flooring parameter β is set to 0.01. These parameters have been determined empirically.[3]

4.6.2 The Generalized Subspace Method

The generalized subspace method proposed by Hu and Loizou [40] generalizes the subspace method by Ephraim and Van Trees [17] – proposed for white noise – to colored noise. Ephraim and Van Trees suggested to perform eigenvalue decomposition on the signal covariance matrix to decompose the vector space of the noisy signal into a signal and a noise subspace. To enhance the signal, the components in the signal subspace are modified by a gain function while the components in the noise subspace are set to zero as they correspond to noise.

The subspace methods of Hu and Loizou and Ephraim and Van Trees are based on the assumption that the clean signal in time domain can be modeled as

$$\mathbf{x} = \boldsymbol{\Psi}\mathbf{s}, \tag{4.6}$$

where $\boldsymbol{\Psi}$ is a $K \times M$ matrix with rank $M(M < K)$ and $\mathbf{s}$ is an $M \times 1$ vector. The matrix $\boldsymbol{\Psi}$ contains complex-valued linearly independent basis vectors [17]. The covariance matrix of $\mathbf{x}$ is

$$\mathbf{R_x} = E\{\mathbf{x}\mathbf{x}^T\} = \boldsymbol{\Psi}\mathbf{R_s}\boldsymbol{\Psi}^T, \tag{4.7}$$

where $\mathbf{R_s}$ is the covariance matrix of $\mathbf{s}$. Due to the signal model in (4.6), $\mathbf{R_x}$ is of rank M and has $K - M$ zero eigenvalues. The noise is assumed to be additive and uncorrelated, i.e.,

$$\mathbf{y} = \boldsymbol{\Psi}\mathbf{s} + \mathbf{n} = \mathbf{x} + \mathbf{n}, \tag{4.8}$$

where $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{n}$ are the K-dimensional vectors containing the noisy speech, the clean speech and the noise signal. To estimate the clean speech vector, a linear estimator $\mathbf{H}$ is defined such that the clean signal estimate is

$$\hat{\mathbf{x}} = \mathbf{H}\mathbf{y}, \tag{4.9}$$

where $\mathbf{H}$ is a $K \times K$ matrix. The error signal between estimated and true clean signal is given by

$$\boldsymbol{\epsilon} = \hat{\mathbf{x}} - \mathbf{x} = \mathbf{H}(\mathbf{x} + \mathbf{n}) - \mathbf{x} = (\mathbf{H} - \mathbf{I})\mathbf{x} + \mathbf{H}\mathbf{n} = \boldsymbol{\epsilon}_{\mathbf{x}} + \boldsymbol{\epsilon}_{\mathbf{n}}, \tag{4.10}$$

[3] The results in the published articles are partly based on different parameter settings. Spectral subtraction is executed on the magnitude while the other parameters are left as provided by [2]. These settings are preferred due to minor occurrence of musical noise, that, however, comes at the price of reduced overall quality.

where $\boldsymbol{\epsilon}$ is composed of the error occurring from speech distortion $\boldsymbol{\epsilon}_{\mathbf{x}}$ and the error occurring from residual noise $\boldsymbol{\epsilon}_{\mathbf{n}}$. The optimal linear estimator based on constraints in the time-domain can be found by minimizing the optimization problem

$$\begin{array}{c} \min_H \bar{\epsilon_{\mathbf{x}}^2} \\ \text{subject to:} \quad \frac{1}{K}\bar{\epsilon_{\mathbf{n}}^2} \leq \tau^2 \quad , \end{array} \tag{4.11}$$

where τ^2 is a positive constant and $\bar{\epsilon_{\mathbf{x}}^2}$ and $\bar{\epsilon_{\mathbf{n}}^2}$ are energies of signal distortion and residual noise, defined as

$$\bar{\epsilon_{\mathbf{x}}^2} = E\{\boldsymbol{\epsilon}_{\mathbf{x}}^T\boldsymbol{\epsilon}_{\mathbf{x}}\} = \mathrm{tr}(E\{[\boldsymbol{\epsilon}_{\mathbf{x}}\boldsymbol{\epsilon}_{\mathbf{x}}^T]\}) \tag{4.12}$$

$$\bar{\epsilon_{\mathbf{n}}^2} = E\{\boldsymbol{\epsilon}_{\mathbf{n}}^T\boldsymbol{\epsilon}_{\mathbf{n}}\} = \mathrm{tr}(E\{[\boldsymbol{\epsilon}_{\mathbf{n}}\boldsymbol{\epsilon}_{\mathbf{n}}^T]\}). \tag{4.13}$$

Thus, the speech energy is minimized while the noise energy is kept below τ^2. The solution for this problem was proposed by Ephraim and Van Trees and is

$$\mathbf{H}_{opt} = \mathbf{R}_{\mathbf{x}}(\mathbf{R}_{\mathbf{x}} + \mu\mathbf{R}_{\mathbf{n}})^{-1}, \tag{4.14}$$

where $\mathbf{R}_{\mathbf{x}}$ and $\mathbf{R}_{\mathbf{n}}$ are the covariance matrices of clean speech and noise, respectively, and μ is the Lagrange multiplier. Using the eigenvalue decomposition $\mathbf{R}_{\mathbf{x}} = \mathbf{U}\boldsymbol{\Delta}_{\mathbf{x}}\mathbf{U}^T$, (4.14) is equivalent to

$$\mathbf{H}_{opt} = \mathbf{U}\boldsymbol{\Delta}_{\mathbf{x}}(\boldsymbol{\Delta}_{\mathbf{x}} + \mu\mathbf{U}^T\mathbf{R}_{\mathbf{n}}\mathbf{U})^{-1}\mathbf{U}^T, \tag{4.15}$$

where $\mathbf{U}$ is the orthogonal eigenvector matrix and $\boldsymbol{\Delta}_{\mathbf{x}}$ is the diagonal eigenvalue matrix of $\mathbf{R}_{\mathbf{x}}$. In the case of white noise $\mathbf{R}_{\mathbf{n}} = \sigma_{\mathbf{n}}^2\mathbf{I}$, $\mathbf{U}^T\mathbf{R}_{\mathbf{n}}\mathbf{U}$ is diagonal. In the case of colored noise, however, $\mathbf{R}_n$ is not diagonal and $\mathbf{U}^T\mathbf{R}_{\mathbf{n}}\mathbf{U}$ is not diagonal, as the eigenvector matrix $\mathbf{U}$ only diagonalizes $\mathbf{R}_{\mathbf{x}}$. To generalize the subspace approach to colored noise, Hu and Loizou proposed to find an eigenvector matrix $\mathbf{V}$ that jointly diagonalizes $\mathbf{R}_{\mathbf{x}}$ and $\mathbf{R}_{\mathbf{n}}$, such that

$$\begin{aligned} \mathbf{V}^T\mathbf{R}_{\mathbf{x}}\mathbf{V} &= \boldsymbol{\Lambda}_{\mathbf{x}} \\ \mathbf{V}^T\mathbf{R}_{\mathbf{n}}\mathbf{V} &= \mathbf{I}. \end{aligned} \tag{4.16}$$

This can be done by solving the eigenvalue equation

$$\boldsymbol{\Sigma}\mathbf{V} = \mathbf{V}\boldsymbol{\Lambda}_{\mathbf{x}}, \tag{4.17}$$

where $\boldsymbol{\Sigma} = \mathbf{R}_{\mathbf{n}}^{-1}\mathbf{R}_{\mathbf{x}}$ [100]. Note that $\mathbf{V}$ is generally not orthogonal because $\boldsymbol{\Sigma}$ is normally not symmetric. Using the relations in (4.16), the optimal estimation matrix in (4.14) can be rewritten as

$$\begin{aligned} \mathbf{H}_{opt} &= (\mathbf{V}^T)^{-1}\boldsymbol{\Lambda}_{\mathbf{x}}\mathbf{V}^{-1}\left[(\mathbf{V}^T)^{-1}\boldsymbol{\Lambda}_{\mathbf{x}}\mathbf{V}^{-1} + \mu(\mathbf{V}^T)^{-1}\mathbf{V}^{-1}\right]^{-1} \\ &= (\mathbf{V}^T)^{-1}\underbrace{\boldsymbol{\Lambda}_{\mathbf{x}}(\boldsymbol{\Lambda}_{\mathbf{x}} + \mu\mathbf{I})^{-1}}_{\mathbf{G}}\mathbf{V}^T \end{aligned} \tag{4.18}$$

For enhancement, the noisy signal is multiplied by this matrix to find the estimator of the clean signal $\hat{\mathbf{x}} = \mathbf{H}_{opt}\mathbf{y}$. This is equivalent to the following steps: First, the transformation $\mathbf{V}^T$ is applied to the noisy signal $\mathbf{y}$, then, the components of $\mathbf{V}^T\mathbf{y}$ are modified by the gain matrix $\mathbf{G}$ and finally the inverse transformation $(\mathbf{V}^T)^{-1}$ is applied. The gain matrix $\mathbf{G}$ is diagonal and composed of the entries

$$g_{kk} = \begin{cases} \frac{\lambda_{\mathbf{x}k}}{\lambda_{\mathbf{x}k}+\mu}, & k = 1, 2, \ldots, M \\ 0, & k = M+1, ..., K, \end{cases} \tag{4.19}$$

where $\lambda_{\mathbf{x}k}$ is the k^{th} diagonal element of $\mathbf{\Lambda_x}$ and M is the rank of the matrix $\mathbf{\Sigma}$ and the assumed dimension of the signal subspace. The Lagrange multiplier μ controls the tradeoff between residual noise and speech distortion and is set depending on the short-time SNR

$$\mu = \mu_0 - (\mathrm{SNR}_{\mathrm{dB}})/s, \tag{4.20}$$

where μ_0 and s are constants defining the degree of noise suppression. This is similar to spectral subtraction with oversubtraction (cf. (4.5) in Section 4.6.1).

In the transform domain, the energy along an eigenvector is equal to the corresponding eigenvalue. Thus, the SNR can by derived as

$$\mathrm{SNR}_{\mathrm{dB}} = 10 \log \frac{tr(\mathbf{V}^T\mathbf{R_x}\mathbf{V})}{tr(\mathbf{V}^T\mathbf{R_n}\mathbf{V})} = \frac{\sum_{k=1}^{M} \lambda_{\mathbf{x}k}}{K}. \tag{4.21}$$

Based on the assumption that noise and speech are uncorrelated, $\mathbf{R_x}$ is estimated from $\mathbf{R_y}$ - $\mathbf{R_n}$, where $\mathbf{R_n}$ is estimated in speech absent frames.

For our experiments, we empirically set the parameter μ_0 to 5 and s to 6.25. In the implementation provided in [2] $\mu_0 = 4.2$. Increasing the value to 5, however, led to a reduction of musical noise.

4.6.3 The Minimum Mean-Square Error Log-Spectral Amplitude Estimator

Ephraim and Malah [99] proposed an estimator for the short-time spectral amplitude based on the minimization of the mean-square error of the log-magnitude spectra. Distortion measures based on the mean-square error of the log-magnitude spectra have been suggested to be more meaningful than distortion measures based on the mean-square error of magnitude spectra [2]. For instance, low speech signal amplitude values are important for speech intelligibility. Therefore, distortion measures that emphasize on small amplitude values are beneficial [48].

The estimation is based on the same statistical model as the MMSE STSA estimator based on the magnitude (proposed by Ephraim and Malah in [12]). The DFT coefficients of speech $X[k]$ and noise $D[k]$ are modeled as statistically independent Gaussian random variables. The motivation for this model is that each Fourier coefficient is a weighted sum of random variables, i.e., the time samples [12]. Under certain mild conditions, the central limit theorem states that the sum of a set of random variables tends to a Gaussian distribution. Hence, the probability density of the Fourier coefficients is modeled as a Gaussian distribution. The statistical independence assumption of DFT coefficients follows from the fact that the correlation between the DFT coefficients decreases when the analysis window length increases. Although the above assumptions are not always fulfilled in practice, i.e., by using shorter and overlapping windows, the derived methods have proven to be useful in practice [2].

For better readability we further denote the magnitude spectra by $X_k = |X[k]|$. The MMSE log-STSA estimator is derived by minimizing the mean-square error between the log-magnitude spectrum of clean speech X_k and its estimate $\hat{X}_k$, given by

$$E\{(\log X_k - \log \hat{X}_k)^2\}. \tag{4.22}$$

The optimal estimator can be found by computing the expectation value of $\log X_k$ conditioned on the observations of the DFT coefficients of the noisy speech signal $Y[k]$, .i.e.,

$$\log \hat{X}_k = E\{\log X_k | Y[k]\}, \tag{4.23}$$

where $Y[k]$ are the DFT coefficients of the noisy speech signal. Hence, the estimator $\hat{X}_k$ evaluates to

$$\hat{X}_k = \exp(E\{\log X_k | Y[k]\}). \tag{4.24}$$

The computation of $E\{\log X_k | Y[k]\}$ can be realized by using the moment-generating function of $\log X_k$ conditioned on $Y[k]$. Using the statistical model discussed above, the estimator can be derived as

$$\hat{X}_k = \frac{\xi_k}{\xi_k + 1} \exp\left\{ \frac{1}{2} \int_{\nu_k}^{\infty} \frac{e^{-t}}{t} dt \right\} Y_k, \tag{4.25}$$

where ξ_k is the *a priori* SNR, $\nu_k = \frac{\xi_k}{1+\xi_k}\gamma_k$ and γ_k is the *a posteriori* SNR. The a priori SNR is defined as $\xi_k = \frac{\sigma_x^2(k)}{\sigma_d^2(k)}$ and the a posteriori SNR is given by $\gamma_k = \frac{Y_k^2}{\sigma_d^2(k)}$, where $\sigma_d^2(k)$ and $\sigma_x^2(k)$ are the variances of the noise and the clean speech signal, respectively. (For further details on the derivation refer to [99, 2]).

Chapter 5

Results and Discussion

In this chapter, we present the evaluation results of the proposed methods. Evaluation was done using objective quality measures and by means of ASR. In addition, visual inspection of spectrograms and listening to the enhanced utterances was useful to gain further insights about the methods.

Objective evaluation is performed by both the PEASS measures and the PESQ measure (see Section 4.3). The PEASS measures allow to evaluate the signal with respect to four aspects. The PESQ measure has shown a high correlation to the scores assessed by subjective listening tests [79]. However, we experienced that it does not seem to consider the presence of musical noise. In addition, we apply the frequency-weighted segmental SNR.

As a benchmark, the proposed methods are compared to the generalized subspace method [40], to spectral subtraction with oversubtraction and spectral flooring [10], and to the MMSE log-STSA estimator [99].

5.1 Kernel PCA and Pre-Image Iterations

The Figures 5.1 and 5.2 compare the results of kernel PCA with the two different pre-image methods NIP and CO as presented in Section 3.4 and 3.5 and published in [39], and the results of PI as explained in Section 3.6 and proposed in [52]. Experiments were conducted on the *airbone* database corrupted by AWGN at 0, 5, 10, and 15 dB SNR. The SNR is computed on the basis of the overall energy of the clean signal (the ASL is not considered). The parameter settings for the kernel variance c and the regularization parameter η are derived from a development set consisting of one sentence per speaker in each SNR condition, which makes six sentences per SNR. For each SNR, c is set to the same value for all utterances. The choice of a suitable value is based on the performance in terms of the PEASS scores, with main focus on the overall quality. For kernel PCA the values for c are 4, 2.5, 0.5, and 0.25 for 0, 5, 10, and 15 dB, respectively. For PI c is set to 4, 2.5, 0.75, and 0.25 for 0, 5, 10,

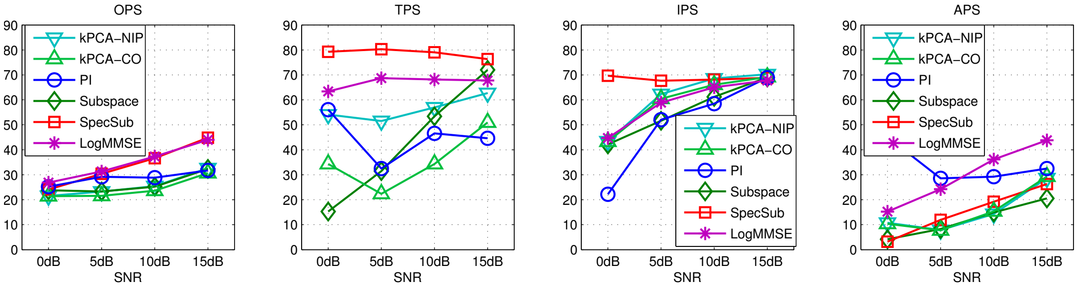


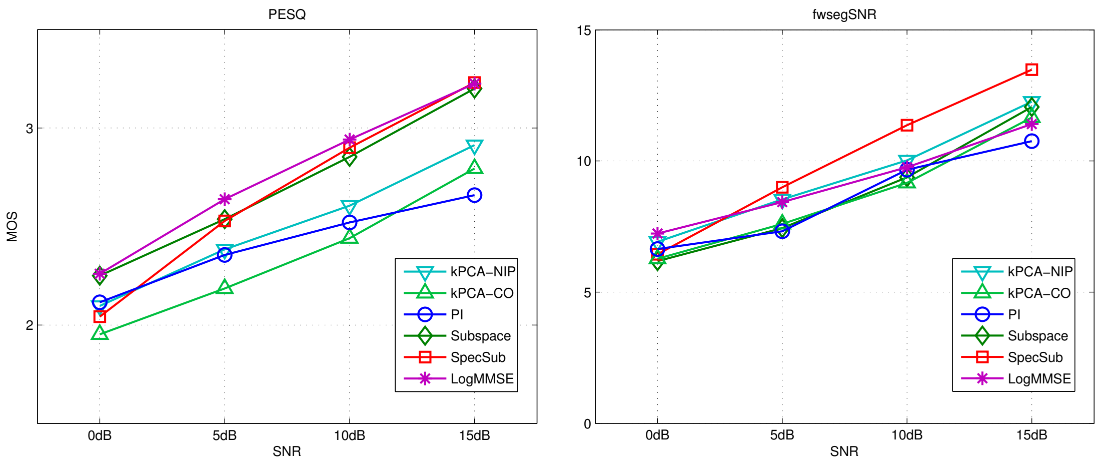
Figure 5.1: Results of kernel PCA with normalized pre-imaging (kPCA-NIP), kernel PCA with combined pre-imaging (kPCA-CO), pre-image iterations (PI), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE log-STSA estimator (LogMMSE) in terms of overall perceptual score (OPS), target perceptual score(TPS), interference perceptual score (IPS), and artifact perceptual score (APS) on the test set of the *airbone* database corrupted by additive white Gaussian noise (AWGN).

and 15 dB, respectively.[1] For PI, regularization is employed and the regularization parameter η is set to 0.5 for all SNR conditions.

Figure 5.1 shows the performance evaluated by the PEASS measures. Kernel PCA with normalized iterative pre-imaging (kPCA-NIP) and the PCA method with combined pre-imaging (kPCA-CO) achieve similar scores, with exception of the TPS. This indicates that not only the disturbing buzz in kPCA-NIP is removed by the combined method kPCA-CO but that also components of the speech are suppressed. PI are slightly superior to the kernel PCA methods and to the generalized subspace method in terms of overall quality (OPS). For low SNRs, the performance of the PI method is similar to spectral subtraction and the MMSE log-STSA estimator, while for high SNRs the other methods are superior. The APS for PI is better than for the other methods in most SNR conditions, indicating that there are few artifacts such as, for instance, musical noise in the case of the generalized subspace method and spectral subtraction. Listening to the files reveals that there is a different type of artifact for PI, namely there is some background noise left around speech components, which is reflected by the rather low IPS.

Figure 5.2 shows the PESQ and the fwsegSNR. In terms of PESQ, the scores for all methods are relatively close, while the reference methods achieve higher scores than the kernel PCA methods and than PI. The presence of musical noise in the recordings enhanced by spectral subtraction and the generalized subspace method is not reflected by the PESQ measure. In terms of fwsegSNR, spectral subtraction performs best, the kPCA-NIP method and the MMSE log-STSA estimator achieve the second best performance depending on the SNR and the other methods achieve little lower SNRs.

[1] The usage of the development set is a difference to the results presented in [39] and [52], where the parameters were set after listening to several example files.



(a) (b)

Figure 5.2: Results of kernel PCA with normalized pre-imaging (kPCA-NIP), kernel PCA with combined pre-imaging (kPCA-CO), pre-image iterations (PI), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE log-STSA estimator (LogMMSE) in terms of the perceptual evaluation of speech quality (PESQ) measure and the frequency-weighted segmental SNR (fwsegSNR) on the test set of the *airbone* database corrupted by AWGN.

5.2 Pre-image Iterations with Automatic Determination of the Kernel Variance for White Noise

To be independent of the knowledge about the SNR, PI were extended by automatic determination of the kernel variance for each individual utterance, as explained in Section 3.7. The results of the PID method presented in this section are published in [51]. AWGN was added at 0, 5, 10, and 15 dB SNR by using the ASL (see Section 4.1.4). Regularization was applied with η equal to 0.25 for 0 dB SNR and 0.75 for the other SNRs. The development set for estimation of the mapping function was extended to two sentences per speaker and SNR condition, in contrast to the PI experiments, where only one sentence was used. In total this amounts to 48 sentences that are derived from 12 clean sentences. The experiments of PI were repeated on the utterances corrupted by noise based on the ASL, which leads to slightly different results in comparison to Section 5.1. For $\mathrm{PI}_{\mathrm{ASL}}$ c was set to 6, 3.5, 0.75, and 0.2 for 0, 5, 10, and 15 dB SNR, respectively, and regularization was applied with η as above.

Figure 5.3 and Figure 5.5 show the PEASS and the PESQ scores. The PID approach performs best in terms of OPS for 0 and 5 dB SNR. For higher SNRs, the MMSE log-STSA estimator and spectral subtraction achieve higher scores. The plots with the standard deviation of the scores in Figure 5.4 show that the standard

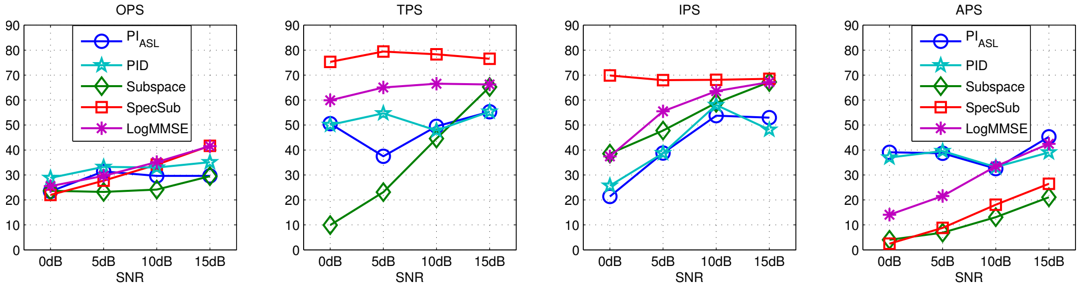


Figure 5.3: Results of pre-image iterations ($\mathrm{PI}_{\mathrm{ASL}}$), pre-image iterations with automatic determination of the kernel variance (PID), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE Log-STSA estimator (logMMSE) in terms of the PEASS scores on the test set of the *airbone* database corrupted by AWGN.

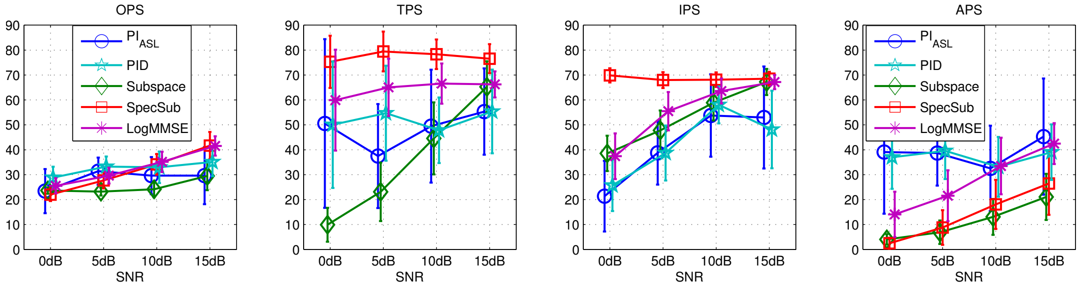
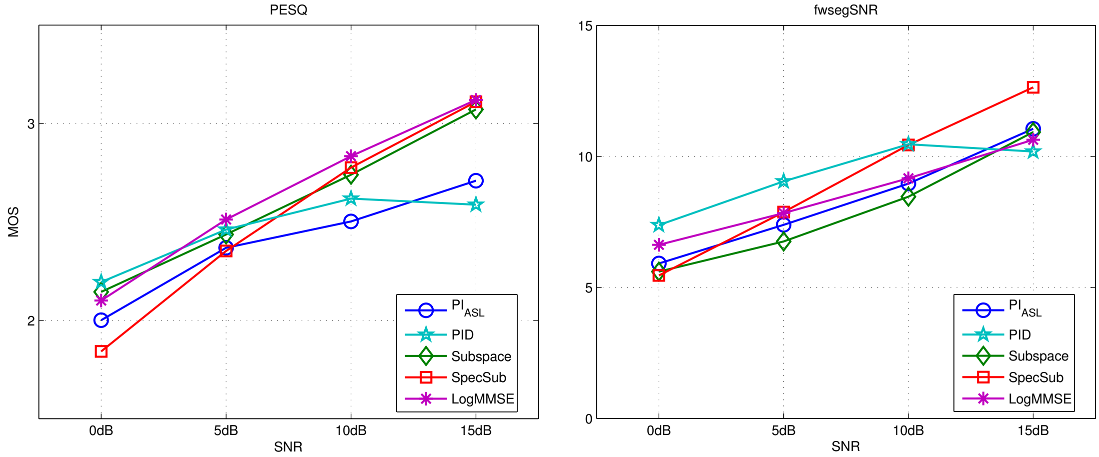


Figure 5.4: The same results as in Figure 5.3 plotted with standard deviation. For better visibility the scores a plotted with a small horizontal offset.



(a) (b)

Figure 5.5: Results of pre-image iterations ($\mathrm{PI}_{\mathrm{ASL}}$), PI with automatic determination of the kernel variance (PID), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE Log-STSA estimator (logMMSE) in terms of the PESQ measure and the fwsegSNR on the test set of the *airbone* database corrupted by AWGN.

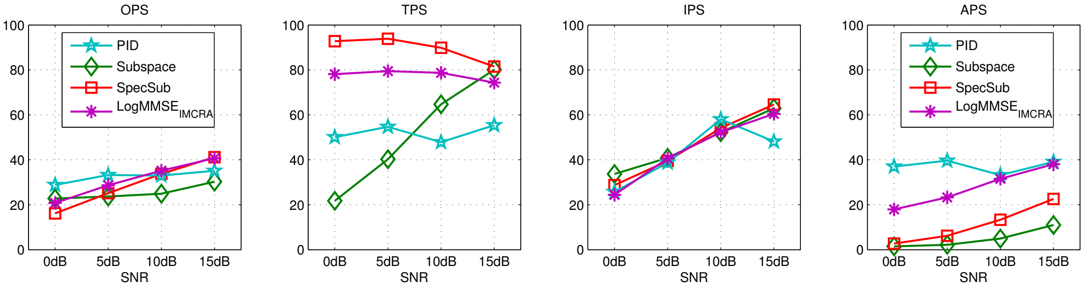


Figure 5.6: Comparison of PI with automatic determination of the kernel variance (PID) and the reference algorithms for parameter settings that effect similar noise reduction measured by the IPS.

deviation of PID is smaller than the standard deviation of PI_{ASL}. This indicates not only an overall improvement but an individual improvement for most sentences.

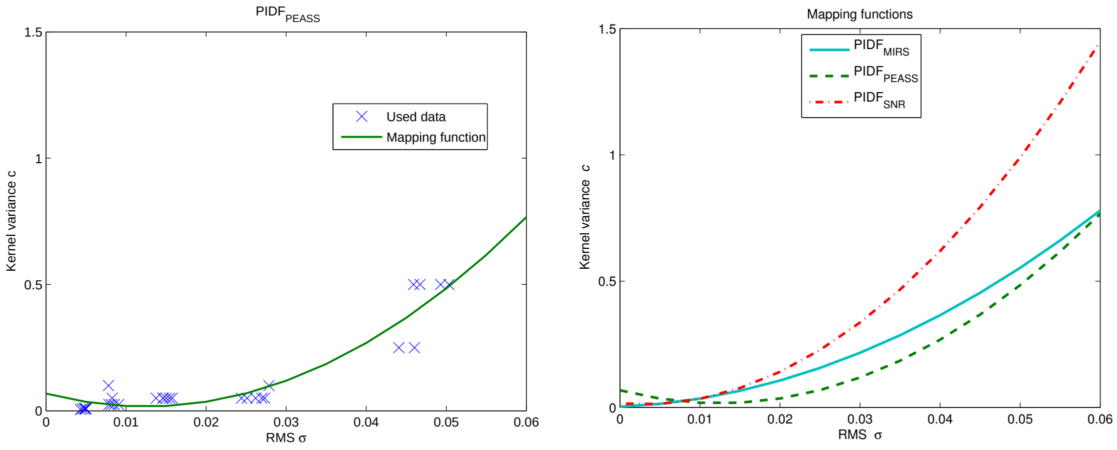
In terms of PESQ all methods are relatively close in performance. PID is superior to PI_{ASL} in all conditions except for 15 dB SNR – presumably the mapping function is not optimal in this condition. In high SNR conditions, the performance of PID is worse than the performance of the reference methods. This also indicates that for high SNRs the mapping function is not optimal and the applied processing is probably too harsh such that speech components are distorted. The suboptimality for high SNRs is as well reflected by the fwsegSNR, for which the performance of PID is good except for 15 dB SNR.

For further analysis, we varied the parameter settings of the reference algorithms in order to achieve a similar noise suppression as with the PID method.[2] This way, the effect of different methods on the speech signal can be compared better. Figure 5.6 shows the PEASS scores of the different enhancement methods with similar IPS. It can be seen that the tendencies of the scores in Figure 5.6 and 5.3 are similar. In terms of TPS, spectral subtraction performs best, however, the OPS and APS are rather low. Listening confirms that the target speaker signal is only mildly distorted, however, there is a lot of residual noise left. In comparison, the generalized subspace method has a lower TPS for low SNRs. Listening and inspection of the spectrograms reveal that this method results in stronger attenuation of high frequency components, although the difference is not as significant as indicated by the difference in TPS.

5.3 Pre-image Iterations with Frequency-Dependent Determination of the Kernel Variance

For colored noise, experiments were conducted on the *Noizeus* and the *airbone* database. The experiments on the *Noizeus* database are more extensive while the

[2] Instead of the basic MMSE log-STSA estimator we use the variant with noise estimation by IMCRA because it results in similar noise suppression as the PID method.



(a) (b)

Figure 5.7: (a) Mapping function of the PIDF$_{\mathrm{PEASS}}$ approach and (b) a comparison of the PIDF$_{\mathrm{MIRS}}$, PIDF$_{\mathrm{PEASS}}$, and PIDF$_{\mathrm{SNR}}$ mapping functions.

experiments on the *airbone* database are rather provided as supplement to the ASR experiments discussed in the subsequent section. In both cases, car noise of the NOISEX-92 database is added at 0, 5, 10, and 15 dB SNR, which is computed with consideration of the ASL. The development set consists of one sentence per speaker and SNR for the *Noizeus* database and two sentences per speaker and SNR for the *airbone* database.

In total, we tested four variants of the PIDF method on data of the *Noizeus* database. The PIDF$_{\mathrm{MIRS}}$ method was published in [51]. It uses a mapping function that is optimized using the criterion S based on the PEASS scores achieved on the development set. For the experiments presented in [51], data corrupted by white noise and filtered by the MIRS filter is used for development. For simplicity, the noise is assumed to be uniformly distributed over the frequency range. This is, however, not valid if the MIRS filter is applied and leads to an underestimation of the noise. The noise variance of the time-domain signal cannot be used as measure for the noise in frequency domain, because in the frequency range where noise is present the noise level is larger. Therefore, the experiments are repeated with data where only the speech data is filtered by the MIRS filter and then the white noise is added. We denote this method by PIDF$_{\mathrm{PEASS}}$. Though the method is more consistent, the noise suppression is rather reduced in comparison to the PIDF$_{\mathrm{MIRS}}$ method.

Two observations can be made when examining the mapping function and the data points used for its estimation in Figure 5.7 (a). First, the distribution of the data points chosen for fitting by the optimization criterion is not optimal: Normally, the mapping function should be monotonously increasing in the used range. The derived mapping function for PIDF$_{\mathrm{PEASS}}$, however, is not monotonous in this range. Apparently, the estimation of data points is poor, i.e., not all values determined for c are suitable. Second, listening to sample utterances reveals that the noise suppres-

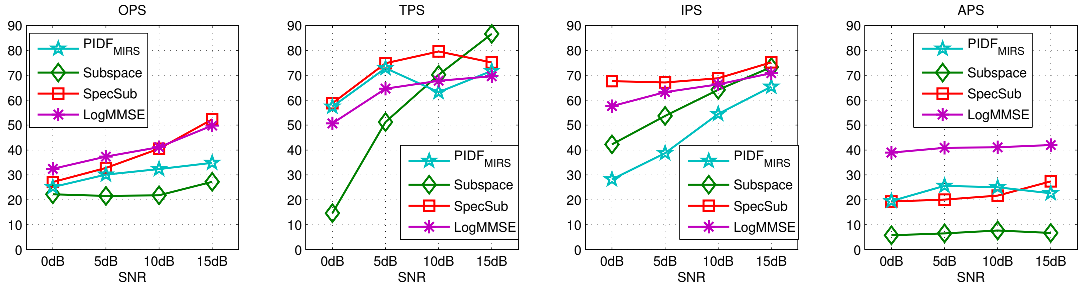
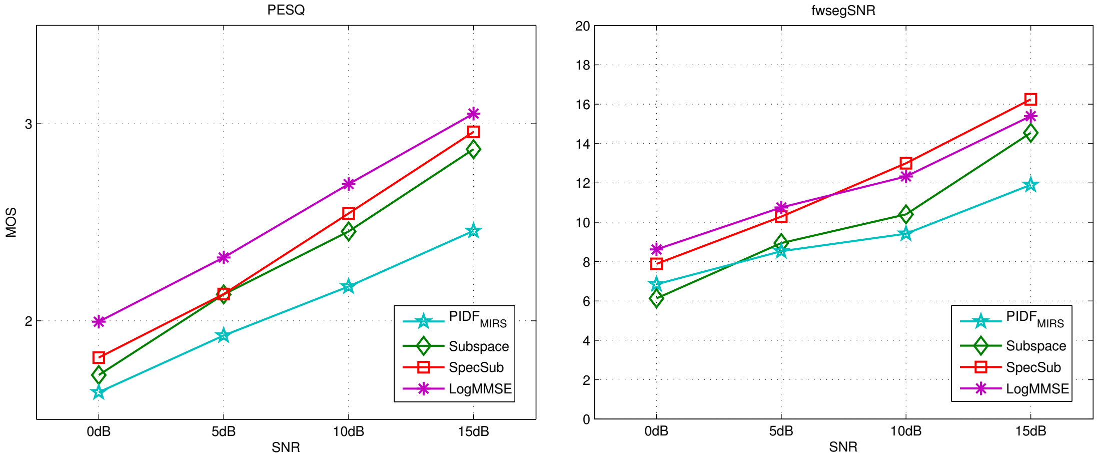


Figure 5.8: Results of pre-image iterations with automatic frequency-dependent determination of the kernel variance (PIDF$_{\mathrm{MIRS}}$), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE log-STSA estimator (LogMMSE) in terms of the PEASS scores on the test set of the *Noizeus* database corrupted by car noise.



(a) (b)

Figure 5.9: Results of PI with automatic frequency-dependent determination of the kernel variance (PIDF$_{\mathrm{MIRS}}$), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE log-STSA estimator (LogMMSE) in terms of the PESQ measure and the fwsegSNR on the test set of the *Noizeus* database corrupted by car noise.

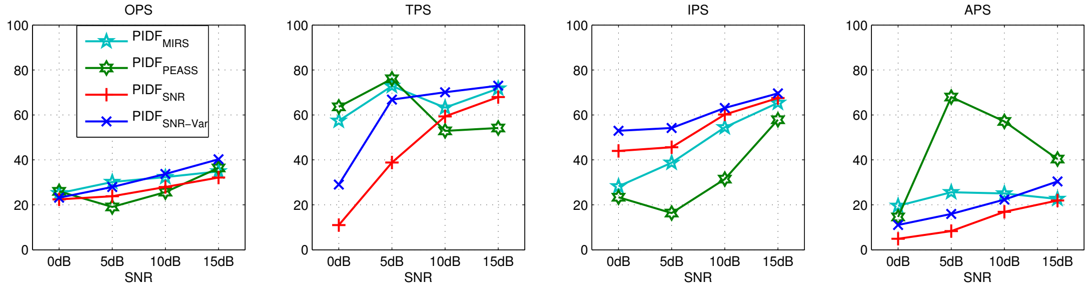


Figure 5.10: Further results for PIDF evaluated on the *Noizeus* database with different mapping function as listed in the text.

sion is limited in comparison to the PIDF$_{\text{MIRS}}$ method – this is consistent with the IPS and the mapping function that mostly results in lower c values for intermediate SNRs as can be seen in Figure 5.7 (b). This causes minor noise suppression in comparison to the mapping function for PIDF$_{\text{MIRS}}$. To achieve better noise suppression a better choice of data points for derivation of the mapping function is required. As the optimization procedure based on the PEASS scores is not satisfying, we tested further measures to determine proper values for c, as explained in Section 3.7.2, and finally used the global SNR. This results in the method denoted by PIDF$_{\text{SNR}}$.

Furthermore, we tested different configurations of the feature extraction. For instance, longer frequency bands lead to stronger noise attenuation as averaging is performed over more samples. For the results denoted by PIDF$_{\text{SNR-Var}}$ we used the mapping function of PIDF$_{\text{SNR}}$ but we processed the signal with frequency bands of 0.4 seconds length and 3 patches height. (For all other experiments the segment length is 0.25 seconds and the number of patches 8.)

In summary, the evaluated variants therefore are:

- **PIDF$_{\text{MIRS}}$** The estimation of c is based on AWGN data filtered by the MIRS filter. For estimation of the mapping function the criterion S derived from the PEASS scores in (3.40) is applied (see Section 3.7.1).

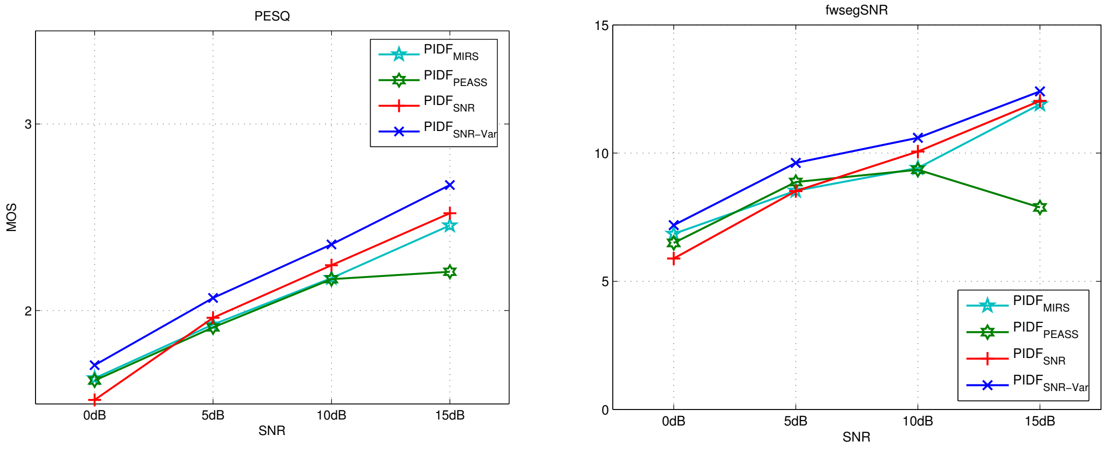
- **PIDF$_{\text{PEASS}}$** The estimation of c is based on data corrupted by AWGN that is not filtered by the MIRS. The criterion S is employed for derivation of the mapping function.

- **PIDF$_{\text{SNR}}$** The estimation of c is based on data corrupted by AWGN not filtered by the MIRS. The global SNR is applied for derivation of the mapping function (see Section 3.7.2).

- **PIDF$_{\text{SNR-Var}}$** The same mapping function as for PIDF$_{\text{SNR}}$ is used, but the configuration for the patch extraction is modified to achieve stronger noise suppression.

The results of PIDF$_{\text{MIRS}}$ in comparison to the generalized subspace method, spectral subtraction and the MMSE log-STSA estimator are shown in Fig 5.8 and 5.9. The overall quality of PIDF$_{\text{MIRS}}$ is better than the subspace method and comparable to spectral subtraction for low SNR conditions. For high SNR conditions, the performance of spectral subtraction is superior. The MMSE log-STSA estimator is superior in terms of OPS. It is interesting to note that PIDF$_{\text{MIRS}}$ achieves consistently high APS scores, however, the IPS indicates a rather limited de-noising performance. This is also reflected by the fwsegSNR, which is lower than for the other methods. In terms of PESQ, the reference methods show superior performance, however, for low SNRs the difference is small.

Figure 5.10 and Figure 5.11 show the comparison of the four PIDF variants. The PIDF$_{\text{MIRS}}$ and the PIDF$_{\text{SNR-Var}}$ method achieves the highest scores for overall quality. The overall performance of the other methods is relatively similar. The



(a) (b)

Figure 5.11: Results of the PIDF methods with different mapping functions in terms of (a) PESQ and (b) global SNR on the test set of the *Noizeus* database.

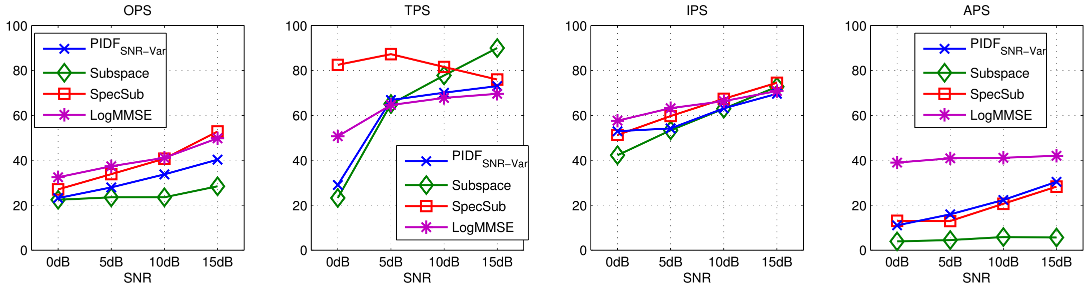


Figure 5.12: Comparison of PI with automatic frequency-dependent determination of the kernel variance ($\mathrm{PIDF}_{\mathrm{SNR\text{-}Var}}$) and the reference algorithms for parameter settings that effect similar noise reduction measured by the IPS.

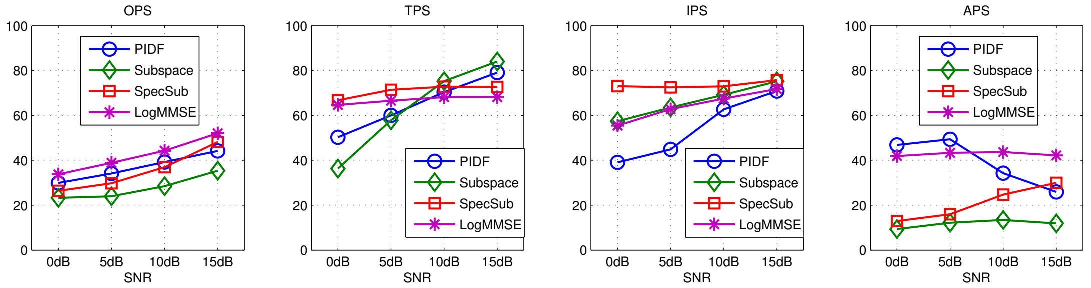
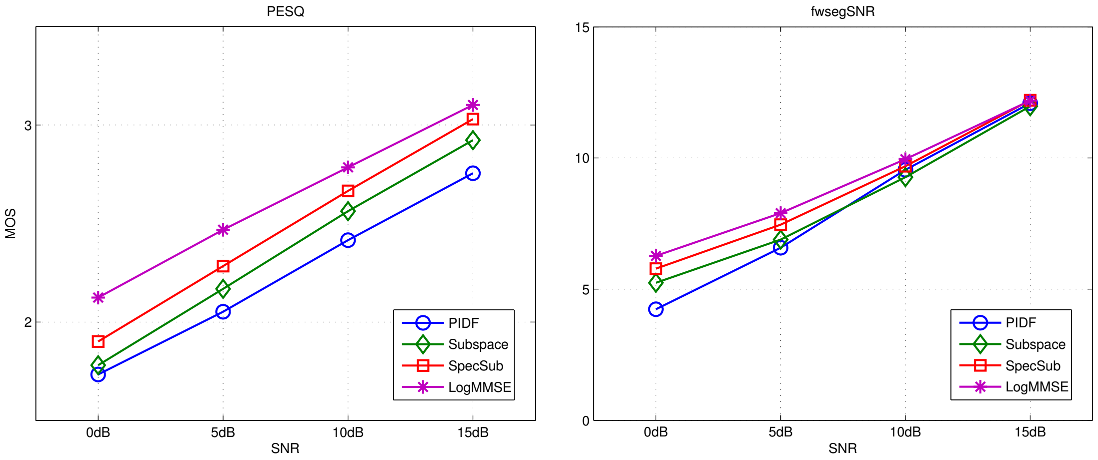


Figure 5.13: Results of pre-image iterations with automatic frequency-dependent determination of the kernel variance (PIDF), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE log-STSA estimator (LogMMSE) in terms of the PEASS scores on the test set of the *airbone* database corrupted by car noise.



(a) (b)

Figure 5.14: Results of pre-image iterations with automatic frequency-dependent determination of the kernel variance (PIDF), the generalized subspace method (Subspace), spectral subtraction (SpecSub), and the MMSE log-STSA estimator (LogMMSE) in terms of the PESQ measure and the fwsegSNR on the test set of the *airbone* database corrupted by car noise.

IPS for $\text{PIDF}_{\text{MIRS}}$ is lower than for PIDF_{SNR} and $\text{PIDF}_{\text{SNR-Var}}$. Listening to the files confirms that the noise suppression with PIDF_{SNR} and $\text{PIDF}_{\text{SNR-Var}}$ is stronger than with the initial $\text{PIDF}_{\text{MIRS}}$ method. These results are consistent with the mapping functions in Figure 5.7 (b). In terms of PESQ, PIDF_{SNR} and $\text{PIDF}_{\text{SNR-Var}}$ perform better than the other methods. Furthermore, the frequency-weighted segmental SNR in Figure 5.11 (b) confirms that the noise suppression with $\text{PIDF}_{\text{SNR-Var}}$ is stronger than with the other variants.

In summary, PIDF_{SNR} and $\text{PIDF}_{\text{SNR-Var}}$ are preferable if strong noise suppression is desired. However, the stronger noise suppression also leads to an attenuation of speech components, especially in the high frequency range. This can reduce the intelligibility.

Figure 5.12 shows the scores of the reference algorithms with a parametrization to effect a similar IPS as achieved by $\text{PIDF}_{\text{SNR-Var}}$. As for the experiment with the PID method, the tendencies of the scores in Figure 5.12 and 5.8 are similar. In comparison to the results of PID with white noise, the performance of the PIDF method is lower in reference to the other methods. This suggests that the generalization to colored noise is not optimal and probably can be further improved.

Figure 5.13 and 5.14 show the evaluation results of the PIDF method on the *airbone* database, where the mapping function is based on the critertion S derived from the PEASS measures (see Section 3.7.1). Note, however, that in contrast to the *Noizeus* database the MIRS filter is not applied on the speech data. In terms of OPS the PIDF method competes with spectral subtraction, while the MMSE log-STSA performs better and the generalized subspace method worse. Note the good APS for PIDF in low SNR conditions which is of special interest for evaluation by ASR. In terms of PESQ and fwsegSNR the performance of all methods is in a similar range, with the PIDF achieving lower scores than the other methods and the MMSE log-STSA achieving the highest scores.

5.4 Musical Noise Suppression

For musical noise suppression we propose two methods. Both derive a mask to suppress musical noise in enhanced recordings. In the first method, a continuous mask is derived by executing PI on the noisy signal. In the second method, PI are performed on the enhanced recording and a binary mask is derived. The methods have been published in [64] and [65].

For execution of PI on the enhanced signal, the two thresholds in (3.43) to derive the mask are set as follows: The lower threshold a is fixed to 1.5, as there are few iteration counts in this range and only the interior of the speech region is affected, which is properly treated by the closing operation anyway. For the upper threshold b, several values are tested on the development set. The one providing the best tradeoff between OPS and APS is taken, ensuring good quality as well as good musical noise suppression. Figure 5.15 shows the results on the development set in four noise conditions when the threshold is varied from 3 to 5. The final values 4, 4, 4.25, and 4.5 are chosen for the noise conditions of 0, 5, 10, and 15 dB SNR, respectively. For these values, the APS is maximized and good artifact suppression, i.e. musical noise suppression, is achieved, while the overall quality is still in the upper range (or maximized as well).

Figure 5.16 shows the results of the musical noise suppression as post-processing step with a continuous mask (CM-MNS) and a binary mask (BM-MNS). Musical noise is poorly reflected by the PESQ measure, therefore we use the PEASS measures to judge the amount of artifacts. The enhanced files, on which the musical noise suppression is applied, have been enhanced by the generalized subspace method. The scores on the data after enhancement are shown as a benchmark. In addition the results after application of the continuous mask on the noisy speech signal are presented (CM-Noisy). Interestingly, the sole application of the mask on the noisy data results in the highest ratings for the overall score. One possibly explanation is that this introduces only few artifacts, as indicated by the APS. However, it has to be noted that the noise suppression is weaker in terms of IPS than for the compared methods.

Regarding the overall quality, the scores before and after post-processing by CM-MNS and BM-MNS are in the same range, while BM-MNS results in slightly higher scores. To judge the effectiveness of the post-processing, we use the APS and the IPS, as musical noise can be seen as an artifact that is caused by the background noise. Indeed both the APS and the IPS improve after post-processing, so the measures reflect that the musical noise is effectively reduced.

Figure 5.17 (a) shows the results when the PESQ measure is used for evaluation.

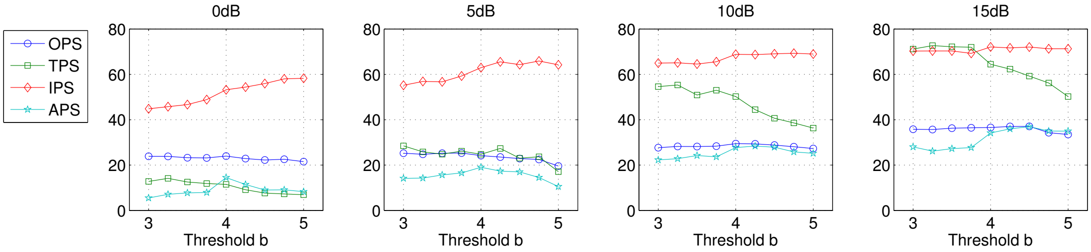


Figure 5.15: Overall perceptual score (OPS), target perceptual score (TPS), interference perceptual score (IPS), and artifact perceptual score (APS) computed from the development set of the *airbone* database for different values of the upper threshold b in different SNR conditions. For the final experiments the threshold maximizing the APS was chosen.

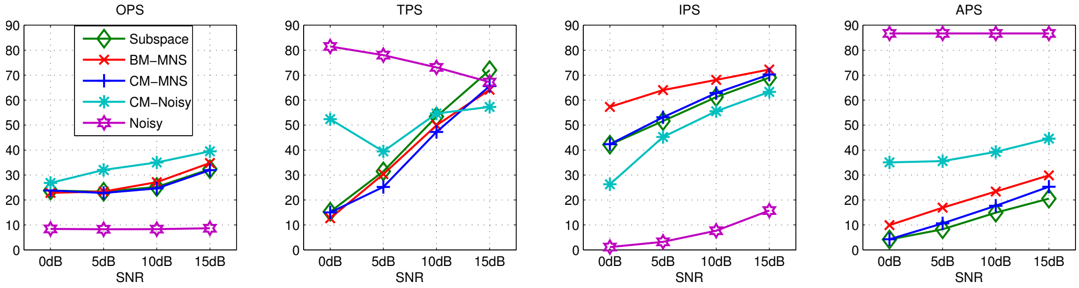


Figure 5.16: Results of musical noise suppression by a binary mask (BM-MNS) and a continuous mask (CM-MNS), compared to the generalized subspace method (Subspace), the noisy signal filtered by the continuous mask (CM-Noisy), and the noisy signal (Noisy) in terms of the PEASS scores on the test set of the *airbone* database corrupted by AWGN.

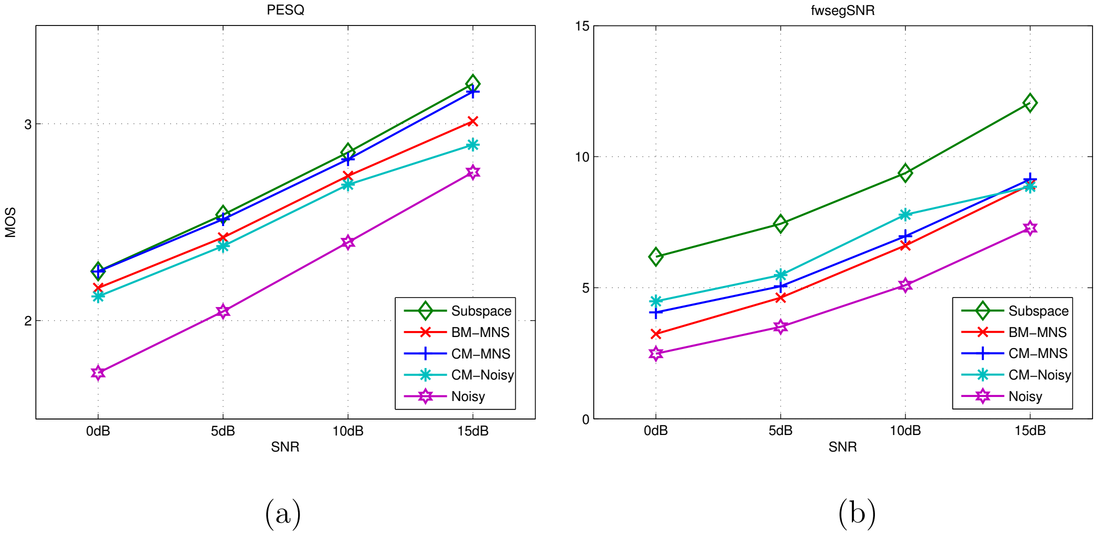


Figure 5.17: Results of musical noise suppression by a binary mask (BM-MNS) and a continuous mask (CM-MNS), compared to the generalized subspace method (Subspace), the noisy signal filtered by the continuous mask (CM-Noisy), and the noisy signal (Noisy) in terms of PESQ and fwsegSNR on the test set of the *airbone* database corrupted by AWGN.

The scores achieved with the PESQ do not agree with the results achieved with the OPS. The performance achieved by simple application of the mask on the noisy signal is lowest. The scores achieved with the subspace method are similar to the scores of CM-MNS, while the performance of BM-MNS is lower. The lower score for BM-MNS can be explained by the fact that it is more unnatural to use a binary mask and that the regions where musical noise is reduced are over-attenuated. The lower performance after MNS indicates that the PESQ measure is no sensitive to musical noise. For reference, the score computed for the unprocessed noisy data is shown. All methods achieve a performance gain in comparison to the unprocessed data (Noisy).

Figure 5.17 (b) shows the results in terms of the frequency-weighted segmental SNR. In all noise conditions, the SNR of the enhanced data is larger than the SNR of the noisy data. The SNR of the subspace method is highest, while the SNR after post-processing by BM-MNS and CM-MNS are lower. One possible reason for this result is that the energy after MNS is lower than the energy in the clean signal, which negatively affects the SNR.

5.5 Automatic Speech Recognition Results[3]

To evaluate speech enhancement in other scenarios than the improvement of perceptual quality, we tested the performance of ASR after enhancement of noise-contaminated data. As explained in Section 4.5, the recognition system was trained on data different from the test data. For training, data of the BAS database was used, while recognition was performed on data of the *airbone* database. PI and PID were evaluated on data corrupted by AWGN, and PIDF was tested on data contaminated by car noise. All experiments were performed for 0, 5, 10, and 15 dB SNR. As a benchmark, the results achieved after enhancement by the generalized subspace method and by spectral subtraction are presented.

For evaluation, we computed the word accuracy (WAcc) in percent achieved on the noisy, enhanced and clean data. The word accuracy is defined as

$$\text{WAcc} = \frac{N - S - D - I}{N} \times 100\%, \tag{5.1}$$

where N is the number of words, S is the number of substitutions, D is the number of deletions and I is the number of insertions [97].

In addition to the WAcc, we evaluated if the performance difference between the PI methods and the compared methods is statistically significant. We use a *matched pairs test* as recommended in [101]. This test is suitable to test the significance of ASR results on speech segments that are statistically independent, i.e., an error in one segment is not influenced by an error in a preceding segment. This is the case for the experiments on the *airbone* database, as we test utterances independent from each other, while errors within one utterance can cause more errors due to

[3] The results presented in this section are based on joint work with Juan A. Morales Cordovilla.

Condition	0 dB	5 dB	10 dB	15 dB	Average
Noisy	3.52	20.93	48.70	73.15	36.58
PI [52]	32.59	58.15	68.52	72.78	58.01
Subspace	2.59	8.33	24.44	54.07	22.36
Subspace$_{\text{MNS}}$	19.63	39.63	54.07	75.56	47.22
SpecSub	35.56	63.15	78.89	88.89	66.62
LogMMSE	47.41	63.70	80.37	90.74	70.56
Clean			97.78		

Table 5.1: Word accuracy (WAcc) in percent achieved on the noisy data, after enhancement (i) by pre-image iterations (PI) as proposed in Section 3.6 and in [52], (ii) by the generalized subspace method, (iii) by the generalized subspace method post-processed by BM-MNS as explained in Section 3.8.2, and (iv) by spectral subtraction (SpecSub), evaluated on the test set of the *airbone* database corrupted by AWGN at 0, 5, 10, and 15 dB SNR. The SNR computation is done without consideration of the ASL.

PI	0 dB	5 dB	10 dB	15 dB
Noisy	*	*	*	
Subspace	*	*	*	*
Subspace$_{\text{MNS}}$	*	*	*	
SpecSub			-	-
LogMMSE	-		-	-

Table 5.2: Results of the statistical significance test between PI and the compared methods for the WAcc in Table 5.1. The asterisk indicates a significantly better performance of PI with a significance level of 0.01, while the minus sign indicates a lower performance.

the restricted grammar. For instance, in each sentence of the database the first element is a name. If this name is not recognized, the recognizer will try to match the remaining words to a name and this results in wrong recognition. The matched pairs test is based on the pair-wise comparison of the recognition rates on the same utterance processed by two algorithms. The difference of errors is computed for each pair and the mean of differences is tested with respect to equality to zero. A mean different from zero indicates a statistical difference of the WAcc of two algorithms. For all evaluations, we employ a significance level of 0.01.

Table 5.1 and 5.3 show the results of PI and PID for AWGN. Table 5.5 shows the results of PIDF for colored noise as described in Section 3.7.2 with derivation of the mapping function for the *airbone* database based on the PEASS scores (see Section 3.7.1). Table 5.2, 5.4, and 5.6 show the results of the significance test between the PI methods and the compared methods for the reported WAcc.

The WAcc for the noisy data clearly states that the recognizer performance suffers

Condition	0 dB	5 dB	10 dB	15 dB	Average
Noisy	0.00	15.56	38.89	65.56	30.00
PI_{ASL}	27.22	53.89	68.33	72.59	57.15
PID [51]	35.93	58.70	72.22	77.59	61.11
Subspace	2.59	4.63	16.30	42.96	16.62
SpecSub	25.74	53.15	73.89	85.56	59.59
LogMMSE	37.78	58.15	74.63	89.07	64.91
Clean			97.78		

Table 5.3: WAcc achieved on the noisy data, after enhancement (i), by PI evaluated on noise-contaminated data based on the ASL, (ii) by PI with automatic determination of the kernel variance (PID) as proposed in Section 3.7.1 and in [51], (iii) by the generalized subspace method, and (iv) by spectral subtraction (SpecSub), evaluated on the test set of the *airbone* database corrupted by AWGN at 0, 5, 10, and 15 dB SNR. The SNR computation is based on the ASL.

PID	0 dB	5 dB	10 dB	15 dB
Noisy	*	*	*	*
PI_{ASL}	*			
Subspace	*	*	*	*
SpecSub	*			-
LogMMSE	-		-	-

Table 5.4: Results of the statistical significance test between PID and the compared methods for the WAcc in Table 5.3. The asterisk indicates a significantly better performance of PID with a significance level of 0.01.

from the noise contamination. The PI methods increase the WAcc in comparison to the noisy data, except for PI in the 15 dB SNR case in Table 5.1, where the enhancement algorithm probably causes artifacts while the SNR is relatively high and the WAcc for noisy data is relatively good.

The results achieved by PI in Table 5.1 show a performance superior to the generalized subspace method. In comparison to spectral subtraction and the MMSE log-STSA estimator the performance is similar or worse. Table 5.2 shows that the WAcc of PI is significantly higher than the WAcc for the noisy signal and the generalized subspace method for all noise conditions except 15 dB SNR. Spectral subtraction is significantly better than PI at high SNR conditions and the MMSE log-STSA estimator is significantly better in almost all conditions. The WAcc of PID and PIDF in Table 5.3 and 5.5 is in all cases superior to the WAcc of the generalized subspace method, similar to the WAcc of spectral subtraction and lower than the WAcc of the MMSE log-STSA estimator. The superior performance is significant for the generalized subspace method, for spectral subtraction at 0 dB SNR and the

Condition	0 dB	5 dB	10 dB	15 dB	Average
Noisy	1.30	25.93	62.78	85.19	43.80
PIDF [51]	34.95	62.04	81.48	89.26	66.93
Subspace	8.52	27.04	66.85	81.48	45.97
SpecSub	29.26	61.11	79.26	90.74	65.23
LogMMSE	52.78	75.74	86.11	94.07	77.17
Clean			97.78		

Table 5.5: WAcc achieved on the noisy data, after enhancement (i) by PI with automatic frequency-dependent determination of the kernel variance (PIDF) as proposed in Section 3.7.2 and in [51], (ii) by the generalized subspace method, and (iii) by spectral subtraction, evaluated on the test set of the *airbone* database corrupted by car noise at 0, 5, 10, and 15 dB SNR. The SNR computation is based on the ASL.

PIDF	0 dB	5 dB	10 dB	15 dB
Noisy	*	*	*	
Subspace	*	*	*	*
SpecSub	*			
LogMMSE	-	-	-	-

Table 5.6: Results of the statistical significance test between PIDF and the compared methods for the WAcc in Table 5.5. The asterisk indicates a significantly better performance of PIDF with a significance level of 0.01.

noisy data except for 15 dB SNR. The comparison of PI_{ASL} to PID in Table 5.3 reveals that the PID method always achieves higher word accuracies.[4] This confirms that the automatic determination of the kernel variance is preferable over using a fixed value for one noise condition. The results for the experiments with car noise in Table 5.5 show that this type of noise is less harmful to the performance of the recognizer. This can be explained by the fact that the noise energy is concentrated below 1kHz, where the speech components are relatively strong and the distortion by the noise therefore is limited.

The good performance of the PI methods in terms of WAcc is a substantial difference to the results of objective quality measures such as PESQ, where the scores of the reference methods are rather higher. One reason for the good performance is presumably, that spectral subtraction and the generalized subspace method are prone to musical noise – especially for the experiments with white noise. In contrast, PI and the MMSE log-STSA estimator do not create such artifacts. The drawback of PI is rather that the high-frequency components may be affected by attenuation. This is a problem if the recordings are evaluated perceptually or by objective qual-

[4] This is statistically significant with a significance level of 0.05.

ity measures, because this affects the speech quality. The ASR system, however, obviously is relatively robust to these degradations.

To test the hypothesis that musical noise is problematic for the speech recognizer we further evaluated the WAcc on the data enhanced by the generalized subspace method and subsequently post-processed by the MNS method proposed in Section 3.8.2. The results are included in Table 5.1 and denoted as Subspace$_{\text{MNS}}$. Indeed, the WAcc is much better after the MNS and the performance difference is significant. Hence, the musical noise is a problem for the recognizer and speech enhancement methods introducing too many artifacts may be counterproductive, as shown for the generalized subspace method.

Finally, the high WAcc on clean data suggests that the recognizer trained on the BAS database generalizes well to the test data of the *airbone* database, although the speakers have different accents (German and the Austrian variety of German) and the vocabulary is not entirely the same.

Chapter 6

Conclusion and Future Work

Speech enhancement is a wide field of research. Since the 1970ies many methods to improve the quality of speech have been proposed. In this work, we propose methods inspired by machine learning techniques and show relations to image de-noising methods. Their power to enhance the quality of noise-corrupted speech is demonstrated. First, we investigate kernel PCA – the non-linear extension to PCA – which has already been successfully applied for image de-noising. Kernel PCA includes an implicit transformation of data samples to the so-called feature space, where the data samples are processed. After processing, a transformation of the processed samples back to input space is necessary. The samples in input space corresponding to the processed samples in feature space are called pre-images. Due to non-linear transformations these pre-images can often only be approximated. Commonly, finding the pre-image is referred to as the pre-image problem.

Experimental results show evidence that for the iterative pre-image methods the weighting factor derived from the projection of kernel PCA only contributes little to de-noising. Therefore, we simplify kernel PCA and an iterative pre-image method and derive the so-called *pre-image iterations* (PI) for noise reduction.

The feature vectors for kernel PCA and PI are complex-valued and extracted from the sequence of short-time Fourier transforms of the speech signal. The de-noising relies on forming linear combinations of noisy feature vectors that are weighted according to their similarity. The similarity is measured by a Gaussian kernel. This method exhibits similarities to non-local filtering and non-local means, which have been used for image de-noising, and to non-local diffusion filters recently applied in speech enhancement for suppression of transient noise. The major difference to these methods, however, is that we apply processing on the complex-valued feature vectors.

Both, kernel PCA and PI, depend on the kernel variance as tuning parameter, which influences the degree of de-noising. Therefore, the performance crucially depends on the setting of the kernel variance. In other words, the tradeoff between de-noising and the possible distortion of speech components, which is inherent to any speech enhancement application, is controlled by the value of the kernel variance.

We therefore generalized the pre-image iteration method by automatic determination of the kernel variance for white noise (PID) and with frequency-dependent determination for colored noise (PIDF). In these modified methods, the noise is estimated from the beginning of a recording which is assumed to be free of speech. Then, the noise estimate is used to derive a suitable value for the kernel variance from a mapping function learned from development data.

Furthermore, analysis of PI shows that information derived from the convergence behavior can be used to discriminate between speech and non-speech regions in time-frequency representations. We use this information to apply musical noise suppression on speech utterances previously enhanced by the generalized subspace method.

In speech enhancement, performance evaluation is still an issue of many open research questions. Several standards exist to assess the speech quality, for instance, for speech coding. In speech enhancement evaluation is more tricky as two aspects have to be covered: (i) background noise is reduced, which results in a quality gain if it is done properly. (ii) any speech enhancement algorithm may also affect the speech components and possibly distort them. Hence, the more the noise is reduced, the higher the probability that speech is degraded. Therefore, the evaluation has to consider both the performance gain by noise reduction and the performance loss by speech distortion originating from noise reduction. In addition, it is problematic if the noise reduction algorithm suppresses noise inconsistently which causes artifacts such as musical noise.

In general, the perceptual quality evaluation can be done objectively by quality measures or subjectively by listening test. There exists a variety of objective quality measures, which more or less correlate with the outcome of subjective listening tests. We considered the PESQ measure, as it has been shown to have high correlation with subjective ratings. Furthermore we used the measures of the PEASS toolbox, which has recently been proposed in the context of source separation. The PEASS measures provide the advantage that the signal is evaluated with regard to four aspects. Therefore deeper analysis is possible in comparison to the PESQ, which provides only a single score.

The evaluation in terms of PESQ and PEASS shows that the performance of the kernel PCA method and of PI for speech enhancement is comparable to the performance of spectral subtraction and superior to the performance of the generalized subspace method, while the MMSE log-STSA estimator achieves rather higher scores. The performance of PID is superior to the performance of PI. This result makes sense, as the individual setting of the kernel variance for each utterance in PID is more advantageous than one value for all utterances of the same SNR. PIDF result in similar performance as spectral subtraction in low SNRs while the performance is weaker in high SNRs. As these are the less difficult conditions, the algorithms can presumably be improved by refining the approach of finding a suitable value for the kernel variance. The reduction of musical noise artifacts by the proposed musical noise suppression methods is confirmed by the artifact perceptual score of the PEASS measures, while the overall quality remains almost unchanged.

Subjective evaluation – by listening tests – was not performed as the objective measures indicated rather small performance differences. In this case only a large-scale listening test could provide statistical evidence, which was not the scope of this work. Listening to the utterances confirms that noise is suppressed. In contrast to the compared methods, no musical noise occurs but there is residual noise around speech components.

In addition to objective quality evaluation, we tested the effect of enhancing speech for automatic speech recognition. The word accuracies on speech enhanced by the PI methods are superior to the word accuracies achieved on noisy speech. In comparison to the generalized subspace method the rates of PID and PIDF are significantly better while they are competing with spectral subtraction and lower than the rates of the MMSE log-STSA estimator. An explanation for the performance difference is the different type of artifacts: While the generalized subspace method and spectral subtraction suffer from musical noise, PI are rather prone to attenuation of high frequency components, which can to a certain degree be controlled by careful choice of the kernel variance. We tested the conjecture that musical noise is problematic for the speech recognizer by comparing the results of the generalized subspace method to the results achieved after post-processing by one of the proposed musical noise suppression algorithms. Indeed, the word accuracies after musical noise suppression are significantly higher, this confirms that the presence of musical noise is problematic for the speech recognizer. In summary, PI provide moderate improvement in speech quality and considerable improvement of recognition rates for the most noise conditions.

The following points are subject to future work:

- Presently, the application of PI is restricted to stationary noise, as the noise is estimated once at the beginning of the utterance. The method can be generalized to non-stationary noise types, such as babble noise, by adding voice activity detection to update the noise estimate in speech pauses or by an extension with state-of-the-art noise tracking methods. Extension by VAD has already been proposed in [102]. However, it was not used for noise estimation but for the separation of speech and non-speech frames, which were processed differently.

- For PID and PIDF the derivation of the mapping function can further be improved. First, other optimization criteria than the currently applied combination of PEASS scores and SNR possibly make the choice of data points for the derivation of the mapping function more robust. Second, a different algorithm to estimate the noise and more elaborate methods than polynomial curve fitting may improve the result.

- PI are computationally demanding, e.g., an utterance of several seconds needs up to around – depending on the parameter settings – 10 minutes processing time on a standard PC. In order to apply PI as pre-processing for automatic speech recognition a speed up is necessary. Furthermore, computational optimization would allow for processing of longer time segments or, equivalently, frequency bands. The application of PI on longer frequency bands has potential to improve de-noising of speech components, as is is more likely to find similar feature vectors within speech regions than it is with the current configuration.

Bibliography

[1] J. Benesty, M. M. Sondhi, and Y. Huang, Eds., *Springer Handbook of Speech Processing*, Springer, 2008.

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

[3] S. Mika, B. Schölkopf, A. Smola, K.-R. Müller, M. Scholz, and G. Rätsch, "Kernel PCA and de-noising in feature spaces," *Advances in Neural Information Processing Systems 11*, pp. 536–542, 1999.

[4] B. Schölkopf and A. J. Smola, *Learning with Kernels*, MIT Press, 2002.

[5] B. Schölkopf, S. Mika, C. J. C. Burges, P. Knirsch, K.-R. Müller, G. Rätsch, and A. J. Smola, "Input space versus feature space in kernel-based methods," *IEEE Transactions on Neural Networks*, vol. 10, pp. 1000–1017, 1999.

[6] K. I. Kim, M. O. Franz, and B. Schölkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1351–1366, 2005.

[7] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 480–530, 2005.

[8] S. F. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, no. 2, pp. 113–120, 1979.

[9] M. Weiss, E. Aschkenasy, and T. W. Parsons, "Study and development of the INTEL technique for improving speech intelligibility," Tech. Rep., Nicolet Scientific Corporation, 1974.

[10] M. Berouti, M. Schwartz, and J. Makhoul, "Enhancement of speech corrupted by acoustic noise," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 208–211, 1979.

[11] R. McAulay and M. Malpass, "Speech enhancement using a soft-decision noise suppression filter," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 2, pp. 137–145, 1980.

[12] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 6, pp. 1109–1121, 1984.

[13] T. Lotter and P. Vary, "Speech enhancement by MAP spectral amplitude estimation using a super-gaussian speech model," *EURASIP Journal on Applied Signal Processing*, vol. 7, pp. 1110–1126, 2005.

[14] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, 2003.

[15] J. S. Lim and A. V. Oppenheim, "Enhancement and bandwidth compression of noisy speech," *Proceedings of the IEEE*, vol. 67, no. 12, pp. 1586–1604, 1979.

[16] M. Dendrinos, S. Bakamidis, and G. Carayannis, "Speech enhancement from noise: A regenerative approach," *Speech Communication*, vol. 10, no. 1, pp. 45–57, 1991.

[17] Y. Ephraim and H. L. Van Trees, "A signal subspace approach for speech enhancement," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 251–266, 1995.

[18] R. Martin, "Spectral subtraction based on minimum statistics," *European signal processing conference (EUSIPCO)*, pp. 1182–1185, 1994.

[19] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 5, pp. 504–512, 2001.

[20] G. Doblinger, "Computationally efficient speech enhancement by spectral minima tracking in subbands," *Eurospeech*, vol. 2, pp. 1513–1516, 1995.

[21] I. Cohen, "Noise spectrum estimation in adverse environments: Improved minima controlled recursive averaging," *IEEE Transactions on Speech and Audio Processing*, vol. 11, pp. 466–475, 2003.

[22] P. S. K. Hansen, *Signal Subspace Methods for Speech Enhancement*, Ph.D. thesis, Technical University of Denmark, 1997.

[23] Y. Hu, *Subspace and Multitaper Methods for Speech Enhancement*, Ph.D. thesis, University of Texas at Dallas, 2003.

[24] K. Hermus, *Signal Subspace Decompositions for Perceptual Speech and Audio Processing*, Ph.D. thesis, Katholieke Universiteit Leuven, 2004.

[25] T. Takiguchi and Y. Ariki, "PCA-based speech enhancement for distorted speech recognition.," *Journal of Multimedia*, vol. 2, no. 5, pp. 13–18, 2007.

[26] J. J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[27] A. Singer, Y. Shkolnisky, and B. Nadler, "Diffusion interpretation of nonlocal neighborhood filters for signal denoising," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 118–139, Jan. 2009.

[28] R. Talmon, *Supervised Speech Processing Based on Geomatric Analysis*, Ph.D. thesis, Technion – Israel Institute of Technology, 2011.

[29] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[30] C. Leitner, F. Pernkopf, and G. Kubin, "Kernel PCA for speech enhancement," *12th Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 1221–1224, 2011.

[31] I. Jolliffe, *Principal Component Analysis*, Springer, 2002.

[32] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," Tech. Rep., Max Planck Institute for Biological Cybernetics, 1996.

[33] J. T. Kwok and I. W. Tsang, "The pre-image problem in kernel methods," *IEEE Transactions on Neural Networks*, vol. 15, pp. 408–415, 2004.

[34] T. J. Abrahamsen and L. K. Hansen, "Input space regularization stabilizes pre-images for kernel PCA de-noising," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2009.

[35] P. Honeine and C. Richard, "Solving the pre-image problem in kernel machines: A direct method," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2009.

[36] Y. Rathi, S. Dambreville, and A. Tannenbaum, "Statistical shape analysis using kernel PCA," *Proc. of SPIE-IS&T Electronic Imaging*, vol. 6064, pp. 60641B–1–60641B–8, 2006.

[37] W.-S. Zheng, J. Lai, and P. Yuen, "Penalized preimage learning in kernel principal component analysis," *IEEE Transactions on Neural Networks*, vol. 21, no. 4, pp. 551–570, 2010.

[38] Y. Rathi, S. Dambreville, and A. Tannenbaum, "Comparative analysis of kernel methods for statistical shape learning," in *Computer Vision Approaches to Medical Image Analysis*, vol. 4241 of *Lecture Notes in Computer Science*, pp. 96–107. Springer Berlin Heidelberg, 2006.

[39] C. Leitner and F. Pernkopf, "The pre-image problem and kernel PCA for speech enhancement," in *Advances in Nonlinear Speech Processing*, vol. 7015 of *Lecture Notes in Computer Science*, pp. 199–206. Springer, 2011.

[40] Y. Hu and P. C. Loizou, "A generalized subspace approach for enhancing speech corrupted by colored noise," *IEEE Transactions on Speech and Audio Processing*, vol. 11, pp. 334–341, 2003.

[41] L. P. Yaroslavsky, *Digital Picture Processing: An Introduction*, Springer-Verlag, Berlin, 1985.

[42] L. Yaroslavsky and M. Eden, *Fundamentals of Digital Optics: Digital Signal Processing in Optics and Holography*, Birkhäuser Boston, 1996.

[43] R. Talmon, I. Cohen, and S. Gannot, "Transient noise reduction using nonlocal diffusion filters," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1584–1599, 2011.

[44] L. R. Rabiner, *Digital Processing of Speech Signals*, Prentice-Hall, 1978.

[45] T. F. Quatieri, *Discrete-Time Speech Signal Processing*, Prentice Hall PTR, 2002.

[46] U. Zölzer, Ed., *DAFX - Digital Audio Effects*, John Wiley & Sons, 2002.

[47] K. Paliwal, K. Wójcicki, and B. Shannon, "The importance of phase in speech enhancement," *Speech Communication*, vol. 53, pp. 465–494, 2011.

[48] P. Vary and R. Martin, *Digital Speech Transmission*, John Wiley & Sons, 2006.

[49] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[50] E. Moulines and W. Verhelst, "Time-domain and frequency-domain techniques for prosodic modification of speech," *Speech Coding and Synthesis*, pp. 519–555, 1995.

[51] C. Leitner and F. Pernkopf, "Generalization of pre-image iterations for speech enhancement," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7010–7014, 2013.

[52] C. Leitner and F. Pernkopf, "Speech enhancement using pre-image iterations," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4665–4668, 2012.

[53] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[54] D. MacKay, *Information Theory, Inference and Learning Algorithms*, Cambridge University Press, 2003.

[55] D. Barash, "A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 844–847, 2002.

[56] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, pp. 2403—-2418, 2001.

[57] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proceedings of the IEEE*, vol. 69, no. 5, pp. 529–541, 1981.

[58] P. Mowlaee and R. Martin, "On phase importance in parameter estimation for single-channel source separation," *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2012.

[59] P. Mowlaee, R. Saeidi, and R. Martin, "Phase estimation for signal reconstruction in single-channel speech separation," *Annual Conference of the International Speech Communication Association (Interspeech)*, 2012.

[60] D. L. Wang and J. S. Lim, "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, no. 4, pp. 679–681, 1982.

[61] A. Stark, K. Wójcicki, J. Lyons, and K. Paliwal, "Noise-driven short-time phase spectrum compensation procedure for speech enhancement," *Annual Conference of the International Speech Communication Association (Interspeech)*, pp. 549–552, 2008.

[62] K. Wójcicki, M. Milacic, A. Stark, J. Lyons, and K. Paliwal, "Exploiting conjugate symmetry of the short-time fourier spectrum for speech enhancement," *IEEE Signal Processing Letters*, vol. 15, pp. 461–464, 2008.

[63] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–2057, 2011.

[64] C. Leitner and F. Pernkopf, "Musical noise suppression for speech enhancement using pre-image iterations," *International Conference on Systems, Signals and Image Procesing (IWSSIP)*, in Press, 2012.

[65] C. Leitner and F. Pernkopf, "Suppression of musical noise in enhanced speech using pre-image iterations," *20th European Signal Processing Conference (EUSIPCO)*, pp. 478–481, 2012.

[66] Z. Goh, K.-C. Tan, and T. Tan, “Postprocessing method for suppressing musical noise generated by spectral subtraction,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 3, pp. 287–292, 1998.

[67] M. Klein and P. Kabal, “Signal subspace speech enhancement with perceptual post-filtering,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002, vol. 1, pp. 537–540.

[68] T. Esch and P. Vary, “Efficient musical noise suppression for speech enhancement systems,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4409–4412, 2009.

[69] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Pearson Prentice Hall, 2008.

[70] F. Schiel and A. Baumann, “Phondat 1, corpus version 3.4,” 2006.

[71] C. Domes, “Kombiniertes Luft- und Knochenleitungsmikrofon-Headset zur robusten Sprachsignalerfassung,” M.S. thesis, Graz University of Technology, 2009.

[72] Y. Hu and P. C. Loizou, “Subjective evaluation and comparison of speech enhancement algorithms,” *Speech Communication*, vol. 49, pp. 588–601, 2007.

[73] IEEE Subcommitee, “IEEE recommended practice for speech quality measurements,” *IEEE Transactions on Audio and Electroacoustics*, vol. 17, no. 3, pp. 225–246, 1969.

[74] H. Hirsch and D. Pearce, “The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions,” *Proceedings ASR*, pp. 181–188, 2000.

[75] ITU-T, “Objective measurement of active speech level,” *ITU-T Recommendation P.56*, 1993.

[76] S. R. Quackenbush, T. P. Barnwell, and M. A. Clements, *Objective Measures of Speech Quality*, Prentice-Hall, 1988.

[77] W. Voiers, “Interdependencies among measures of speech intelligibility and speech “quality”,” *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 5, pp. 703–705, 1980.

[78] S. Bech and N. Zacharov, *Perceptual Audio Evaluation: Theory, Method and Application*, John Wiley & Sons, 2006.

[79] Y. Hu and P. C. Loizou, “Evaluation of objective quality measures for speech enhancement,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 229–238, 2008.

[80] A. W. Rix, "Perceptual speech quality assessment - a review," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 1056–1059, 2004.

[81] A. W. Rix and M. P. Hollier, "The perceptual analysis measurement system for robust end-to-end speech quality assessment," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 3, pp. 1515–1518, 2000.

[82] J. G. Beerends and J. A. Stemerdink, "A perceptual speech-quality measure based on a psychoacoustic sound representation," *Journal of the Audio Engineering Society*, vol. 42, no. 3, pp. 115–123, 1994.

[83] S. Voran, "Objective estimation of perceived speech quality – Part I: Development of the measuring normalizing block technique," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 371–382, 1999.

[84] ITU-T, "Objective quality measurement of telephone-band (300-3400 Hz) speech codecs," *ITU-T Recommendation P.861*, 1998.

[85] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (PESQ) – a new method for speech quality assessment of telephone networks and codecs," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 749 –752, 2001.

[86] ITU-T, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," *ITU-T Recommendation P.862*, 2000.

[87] H. J. Steeneken and T. Houtgast, "Mutual dependence of the octave-band weights in predicting speech intelligibility," *Speech Communication*, vol. 28, pp. 109–123, 1999.

[88] R. Huber and B. Kollmeier, "PEMO-Q – A new method for objective audio quality assessment using a model of auditory perception," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 1902–1911, 2006.

[89] E. Vincent, H. Sawada, P. Bofill, S. Makino, and J. P. Rosca, "First stereo audio source separation evaluation campaign: data, algorithms and results," *7th International Conference on Independent Component Analysis and Signal Separation (ICA)*, pp. 552–559, 2007.

[90] ITU-R, "General methods for the subjective assessment of sound quality," *ITU-R Recommendation BS.1284-1*, 2003.

[91] ITU-T, "Methods for subjective determination of transmission quality," *ITU-T Recommendation P.800*, 1996.

[92] ITU-T, “Subjective performance assessment of telephone-band and wideband digital codecs,” *ITU-T Recommendation P.830*, 1996.

[93] W. D. Voiers, “Diagnostic acceptability measure for speech communication systems,” *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 204–207, 1977.

[94] B. J. McDermott, “Multidimensional analyses of circuit quality judgments,” *Journal of the Acoustical Society of America*, vol. 45, pp. 774–781, 1968.

[95] ITU-T, “Subjective test methodology for evaluating speech communication systems that include noise suppression algorithm,” *ITU-T Recommendation P.835*, 2003.

[96] M. H. L. Hecker and C. E. Williams, “Choice of reference conditions for speech preference tests,” *Journal of the Acoustical Society of America*, vol. 39, pp. 946–952, 1966.

[97] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, *The HTK Book*, Cambridge University Engineering Department, 2006.

[98] H. G. Hirsch, “Experimental framework for the performance evaluation of speech recognition front-ends of large vocabulary task,” Tech. Rep., STQ AURORA DSR, Working Group, 2002.

[99] Y. Ephraim and D. Malah, “Speech enhancement using a minimum mean-square error log-spectral amplitude estimator,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 33, no. 2, pp. 443 – 445, 1985.

[100] S. R. Searle, *Matrix Algebra Useful for Statistics*, John Wiley & Sons, 1982.

[101] L. Gillick and S. Cox, “Some statistical issues in the comparison of speech recognition algorithms,” *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 532–535, 1989.

[102] C. Leitner and F. Pernkopf, “Extension of pre-image speech de-noising by voice activity detection using a bone conductive microphone,” *International Workshop on Acoustic Signal Enhancement (IWAENC)*, 2012.