Carl von Ossietzky
Universität Oldenburg

Studiengang Diplom-Physik

DIPLOMARBEIT

Titel:

Robust Speech Recognition based on Spectro-Temporal Features

vorgelegt von: Bernd Meyer

Betreuender Gutachter: Prof. Dr. Dr. Birger Kollmeier
Zweiter Gutachter: Prof. Dr.-Ing. Alfred Mertins

Oldenburg, April 2004

Contents

List of Figures

List of Tables

1 Introduction

this sentence was transcribed with
his speech recognition software and chills
the progress as well as problems that
is still present in as our.

1.1 Automatic Speech Recognition (ASR)

The first sentence was transcribed with a speech recognition software and shows the progress as well as problems, that are still present in ASR. Although one can think of many applications where automatic speech recognition would be helpful, ASR systems are usually not utilized in everyday life due to several limitations. Commercial dictation software is available to everyone with a computer and works very well in optimal conditions, which demonstrates that a sub-goal in ASR has already been achieved. On the other hand, in conditions that are not as optimal (e.g. if speech from a foreign speaker with an accent is to be recognized, as in the presented example) performance decreases rapidly below the acceptable level, even when the system is trained on the speaker's voice and a close-talk microphone is used. This shows, that in spite of intense research efforts, the goal of conversational speech recognition by machine is far from being achieved. Some of the main problems in ASR are co-articulation, the high complexity and the large variability of speech, i.e. many realizations exist for the same speech unit, even if it is uttered by the same speaker. These problems have not been completely solved yet, which results in ASR error rates ten times larger compared to the human performance, even in optimal acoustic conditions.

1.2 Robustness of ASR systems

Additional problems emerge when speech is disturbed by convolutive or additive noise, that arise from superposition of speech and noise signals or from disturbances of an electric or acoustic transmission channel, e.g. a telephone channel or a room. The invariance of recognition performance under such disturbances is called robustness.

First systems are available that can compensate for modest amounts of acoustical degradation caused by the effects of unknown noise and unknown linear filtering. Still, the performance of even the best state-of-the-art systems is heavily deteriorated in the mentioned adverse conditions. This is one of the main reasons that prevent automatic speech recognition from being used in everyday situations, so increased robustness is still a very desirable property in ASR.

There exist three different approaches in order to achieve this goal:

Firstly, disturbances can be removed from the speech signal before features that carry speech-relevant information are extracted. There exist a number of methods to deal with additive or convolutive noise (like spectral subtraction, processing with the Ephraim-Malah algorithm or inverse filtering). One of the downsides of such processing is that the application of these techniques produces artifacts in the speech signal, for example, due to wrong estimation of the noise signal.

Another approach is to design a robust feature extraction, where features are as invariant as possible under adverse acoustical conditions. This approach was pursued in our work.

Finally, the classifier can be designed to cope with a large variety of noise signals. This can be achieved by training multiple acoustical models with speech under different noise

conditions. The problem with this approach is the large number of these, that dramatically increase computational cost and demand for memory. Another problem is the automatic selection of the appropriate model in dependence of the actual acoustical situation.

1.3 Scope of this thesis

The goal of this thesis is to increase overall performance and especially robustness of ASR systems by using localized, spectro-temporal filters (LSTFs), from which robust features for ASR systems are calculated. The work is led by the idea of learning certain feature extraction techniques from the biological blueprint, which performs much better than any technical ASR system.

The large gap in performance between normal-hearing native listeners and state-of-the art ASR systems is most evident in adverse acoustic conditions. Furthermore, humans outperform machines by at least an order of magnitude (Lippmann, 1997). Human listeners recognize speech even in very adverse acoustical environments with strong reverberation and interfering sound sources. While many cognitive aspects of speech perception still lie in the dark, there is much progress in the research on signal processing in the more peripheral parts of the (human) auditory system.

In (Kleinschmidt, 2002a) the usage of 2-dimensional Gabor filters for ASR was proposed. These physiologically and psycho-acoustically motivated features employ spectro-temporal information inherent to the speech signal. As a starting point, the properties of LSTF features [1] are evaluated: Compared to other features in ASR, the number of feature vector components of LSTF features is relatively high because of the large number of filters in feature prototype sets and due to concatenation with dynamic features. Therefore, LSTFs were analyzed with respect to the number of features and the necessity of dynamic features needed for robust ASR system performance.

Several methods of improvement for LSTFs are then investigated, for which knowledge from physiology and signal-processing was employed. Spectro-temporal receptive fields exhibit properties, that have not been employed in the original Gabor approach. It was investigated, whether increased robustness can be achieved by taking these findings into account. For these experiments a rather simple classifier and a small vocabulary corpus was used. With a more complex back end, a further evaluation of previously used and existing filter sets was carried out. Spectro-temporal features proved to be very robust compared to cepstral coefficients for a digit-recognition task and in combination with classifier with a rather small number of parameters. It was investigated, whether these results are scalable to a more complex back ends and to other corpora.

Because of their spectro-temporal structure, LSTF features clearly differ from cepstral coefficients, which are the most commonly used features. A further goal was to quantify complementarity of both feature types and to evaluate beneficial effects by combining these.

Additionally, it was investigated what other methods are suitable to increase overall performance with a state-of-the-art recognizer. This includes an analysis of advanced noise suppression methods as well as effects of linear transformations.

[1] In (Kleinschmidt, 2002a), the spectro-temporal features are referred to as Gabor features. However, since in this work more generalized modulation filters are proposed for which the name Gabor filter is no longer adequate, the term localized, spectro-temporal filter (LSTF) is used. Features derived from those filters are called LSTF features.

1.4 Overview

The design of ASR systems in general and feature extraction with localized, spectro-temporal filters in particular are covered in section 2. This includes a description of the physiological motivation for LSTFs and of the automatic feature finding process. Methods of feature space transformations and an overview of Hidden Markov models (HMMs) and the Tandem approach, which combines conventional classifiers with artificial neural networks, are presented as well.

The corpora that have been used to calculate feature prototype sets and to evaluate ASR systems with LSTF features are introduced in section 3.

Section 4 deals with experiments regarding the evaluation and improvement of LSTF features, as well as the design of new filter types, so called fully separable filter functions.

The questions regarding scalability of results, complementarity and overall performance of LSTF features with a state-of-the-art system are discussed in section 5, where a further evaluation of previously used and optimized features was carried out. For these tests, the ASR system `ASPIRIN`, which is used for research at Philips Research Laboratories, Aachen, was used.

The summary and conclusions are presented in section 6 and detailed results and a list of abbreviations are given in section 7.

2 LSTFs, Linear Transformations and Likelihoods

In this section, an overview over the different stages used in ASR systems is given with focus on feature extraction based on localized, spectro-temporal filters (LSTFs). An schematic overview of an ASR system as used in our experiments is presented in Figure 1.

Feature extraction deals with the separation of ASR relevant information and the data that is not needed to transcribe the utterance. Therefore, useful variability that can help to identify a word or sentence should be emphasized, whereas variability characterizing speaker identity, the speaker's emotional state and environmental effects should be neglected.

A two-stage feature extraction process may be used to achieve this: From a waveform, a spectro-temporal representation referred to as primary feature matrix is extracted at a 100 Hz frame rate. From this, secondary features are calculated, which yields a feature vector per time frame. A detailed description of the LSTF feature extraction process is given in subsection 2.1.

Feature vectors may be object to linear or non-linear transformations, that are used to reduce computational load in further processing stages and to improve overall performance by decorrelation of feature vector components. Principal component analysis (PCA) and linear discriminant analysis (LDA) are two transformations, that have been used in our experiments; an overview of these techniques is given in subsection 2.2.

Features are fed to an acoustic model, where knowledge about structure and parameters of relevant linguistic units and their acoustic correlatives is employed. For most of today's ASR systems either Gaussian Mixture Models (GMMs) or artificial neural networks (ANNs) are used as acoustic model.

The output of these models provides the likelihoods or probabilities for different speech sounds (usually phonemes), that are fed to a Hidden Markov Model (HMM) decoder, which searches for the most likely phoneme and word sequence. A GMM-HMM system and a ANN have been successfully combined in the Tandem approach (Hermansky et al., 2000). The functionality of both HMMs as well as Tandem ASR systems are presented in subsection 2.3.

2.1 Localized spectro-temporal features

2.1.1 Psychoacoustical and physiological motivation for LSTFs

Recent findings from a number of physiological experiments in different mammal species showed that a large percentage of neurons in the primary auditory cortex (A1) respond differently to upward- versus downward-moving ripples in the spectrogram of the input (Depireux et al., 2001). Spectro-temporal receptive fields (STRFs) show that individual neurons are sensitive to specific spectro-temporal modulation frequencies in the incoming sound signal.
The STRF is a model representation of excitatory and inhibitory integration area of auditory neurons (Qui et al., 2003). It is proportional to the linear component of its estimated optimal stimulus and describes the spectral and temporal attributes that preferentially activate a neuron. In order to determine the STRF of a neuron, spike-triggered averages are calculated for a series of time-frames extending back in time from the moment of neural activity. This activity may be invoked using complex spectro-temporal stimuli such as checkerboard noise (deCharms et al., 1998) or moving ripples

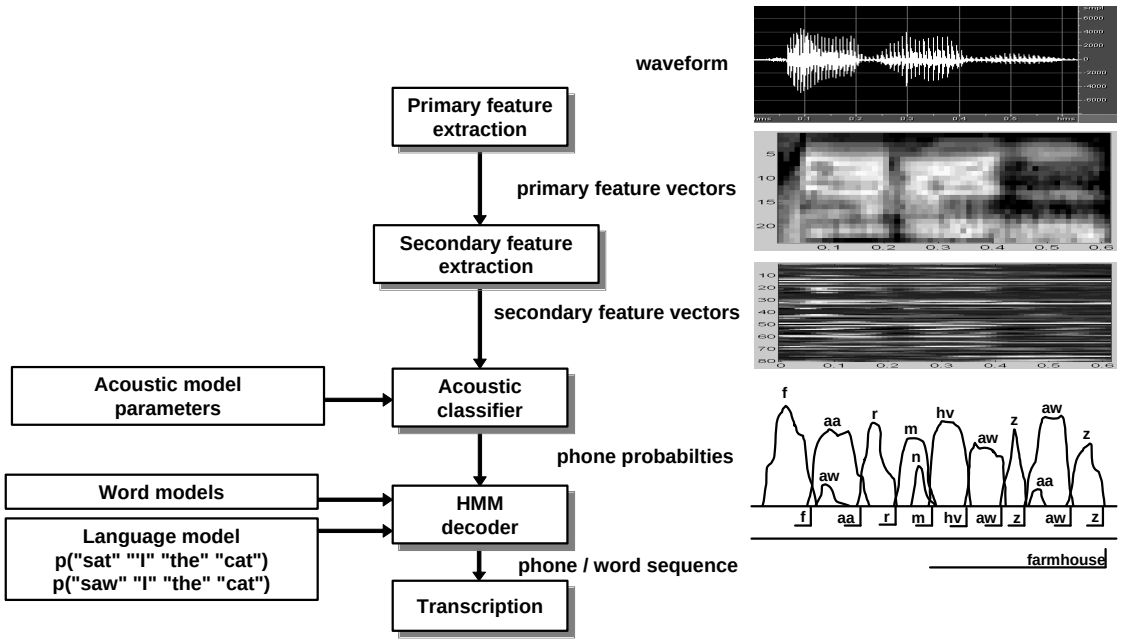

Figure 1: Structure of typical ASR system with the main stages feature extraction, acoustic classification and word- and language modeling. On the right hand side, an example for data representation at different stages is given for the recognition of the word "farmhouses" (adapted from (Ellis and Gelbart, 2004)).

(Schreiner and Calhoun, 1994). Two examples of STRFs that exhibit diagonal structures are depicted in Figure 2.

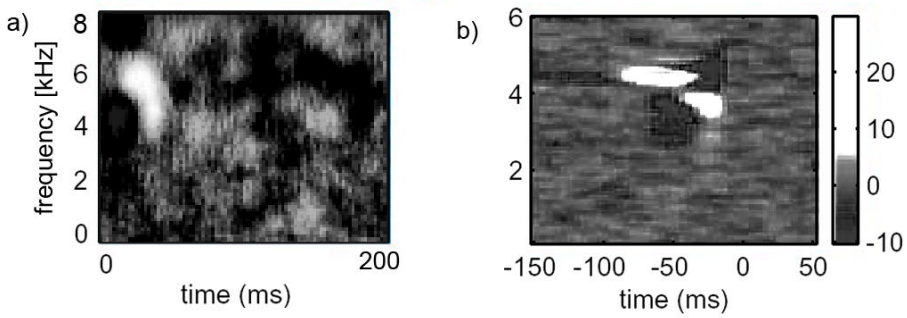

Figure 2: Examples for spectro-temporal receptive fields (STRFs) in time-frequency domain adapted (from (Elhilali et al., 2003) and (deCharms et al., 1998))

The STRFs often clearly exceed one critical band in frequency, have multiple peaks and also show tuning to temporal modulation (see (Schreiner et al., 2000)). Still, the STRF patterns are mainly localized in time and frequency, generally spanning at most 250 ms and one or two octaves, respectively. The center frequency distributions of the linear modulation filter transfer function associated with the STRFs show a broad peak between 4 and 8 Hz in the ferret's A1 and at about 12 Hz in the cat's A1 (Miller et al., 2002).

The neurophysiological data fit well with psychoacoustic experiments on early auditory features: in (Kaernbach, 2000) a psychophysical reverse correlation technique was applied to masking experiments with semi-periodic white noise. The resulting basic auditory feature patterns are distributed in time and frequency and in some cases are comprised of several unconnected parts, very much resembling the STRF of cortical neu-

rons. Often, two neurons show very similar STRFs differing only by a $\pi/2$ phase shift. Two such cells combined provide for a translation-invariant detection of a given modulation pattern within a certain part of the spectro-temporal representation of a speech signal. In the visual cortex, spatio-temporal receptive fields are measured with (moving) orientated grating stimuli. The results very well match two-dimensional Gabor functions (De-Valois and De-Valois, 1990). The use of 2D complex Gabor filters as features for ASR has been proposed earlier and proven to be relatively robust in combination with a simple classifier (Kleinschmidt, 2002a).

Automatic feature selection methods are described in subsection 2.1.2 and the resulting parameter distribution has been shown to remarkably resemble neurophysiological and psychoacoustical data as well as modulation properties of speech (Kleinschmidt, 2003).

This approach of spectro-temporal processing by using localized sinusoids most closely matches the neurobiological data and also incorporates other features as special cases: purely spectral Gabor functions perform sub-band cepstral analysis—modulo the windowing function—and purely temporal ones can resemble temporal patterns (TRAPS) or the relative spectra transformation (RASTA) impulse response and its derivatives (Hermansky, 1998) in terms of temporal extent and filter shape.

Speech is characterized by its fluctuations across time and frequency. The latter reflect the characteristics of the human vocal cords and tract and are commonly exploited in ASR by using short-term spectral representations such as cepstral coefficients. The temporal properties of speech are targeted in ASR by dynamic (delta and delta-delta) features and temporal filtering and feature extraction techniques like RASTA and TRAPS (Hermansky, 1998).

Nevertheless, speech clearly exhibits combined *spectro-temporal* modulations. This is due to intonation, coarticulation and the succession of several phonetic elements, e.g., in a syllable. Formant transitions, for example, result in diagonal features in a spectrogram representation of speech. An example for this is shown in Figure 3. This kind of pattern is explicitly targeted by the feature extraction method used in our experiments.

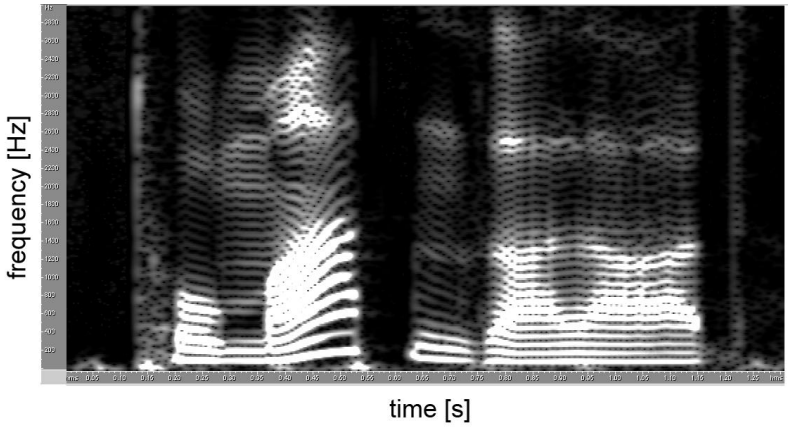

Figure 3: Spectrogram of the utterance "Tomatensalat", where spectro-temporal structures may be identified. Brightness denotes energy.

There are a number of different approaches to achieve spectro-temporal feature extraction for ASR, such as spectro-temporal modulation filtering (Nadeu et al., 2001), and the extension of TRAPS to more than one critical band (Jain and Hermansky, 2003).

Approaches to use artificial neural networks for ASR classify spectral features using temporal context on the order of 10 to 100 ms. Depending on the system, this is part

of the back end as in the connectionist approach (Bourlard and Morgan, 1998) or part of the feature extraction as in the Tandem system that is presented in subsection 2.3.2. None of the above feature extraction techniques combine the advantages of scalable, localized spectro-temporal modulations filter prototypes with an efficient feature set selection algorithm, as it is done in the approach presented here.

The usage of spectro-temporal processing seems to be a general trend in the ASR community. The Hidden Activation TRAPS (HATS) approach as proposed in (Chen et al., 2003) shows remarkable parallels to the LSTF approach: HATS is based on feature extraction according to temporal patterns (TRAPS), that were developed with the same physiological motivation as the Gabor filters (deCharms et al., 1998). In TRAPS, a set of multi-layer perceptrons (MLPs) is trained, with each MLP having as input Mel–scale spectral energy values in one critical band over a long time trajectory of about 1 s. A merger MLP combines the output values of the critical band MLPs. In HATS, instead of combining the values of the output layer, only the hidden values are used after training and the output units are ignored. Extensions of these methods are triband-TRAPS and triband-HATS, where the three adjacent frequency channels are used as input to the MLPs. The usage of multiple frequency bands allows for spectro-temporal processing similar to the LSTF approach. Furthermore, the determination of filters shows also parallels: The input-layer-to-hidden-unit weights in HATS are obtained by discriminatively training, just as the LSTF filter sets. In both cases output values are input to an acoustical classifying MLP.

2.1.2 Feature extraction method

From an input signal, a spectro-temporal representation — the primary feature matrix — is calculated. This representation is processed by a number of 2-D modulation filters. The filtering is performed by correlation over time of each input frequency channel with the corresponding part of the LSTF function (centered on the current frame and desired frequency channel) and a subsequent summation over frequency. This yields one output value per frame per filter and is equivalent to a 2-D correlation of the input representation with the complete filter function and a subsequent selection of the desired frequency channel of the output. Filter outputs are referred to as secondary features. In this study, log mel-spectrograms serve as input features for feature extraction. This was chosen for its widespread use in ASR and because the logarithmic compression and mel-frequency scale might be considered a very simple model of peripheral auditory processing. Any other spectro-temporal representation of speech could be used instead and especially more sophisticated auditory models might be a good choice for future experiments.

The two-dimensional complex Gabor function $G(n, k)$ as proposed in (Kleinschmidt, 2002c) for ASR is defined as the product of a Gaussian envelope $g(n, k)$ and the complex sinusoidal function $s(n, k)$ (c.f. Fig. 2.1.2 a and c). The envelope width is defined by standard deviation values σ_n and σ_k, while the periodicity is defined by the radian frequencies ω_n and ω_k with n and k denoting the time and frequency index, respectively. The two independent parameters ω_n and ω_k allow the Gabor function to be tuned to particular directions of spectro-temporal modulation, including *diagonal* modulations. Further parameters are the centers of mass of the envelope in time and frequency n_0 and k_0. In this notation the Gaussian envelope $g(n, k)$ is defined as

$$g(n,k) = \frac{1}{2\pi\sigma_n\sigma_k} \cdot \exp\left[\frac{-(n-n_0)^2}{2\sigma_n^2} + \frac{-(k-k_0)^2}{2\sigma_k^2}\right] \tag{1}$$

and the complex sinusoid $s(n,k)$ as

$$s(n,k) = \exp\left[i\omega_n(n-n_0) + i\omega_k(k-k_0)\right]. \tag{2}$$

The envelope width is chosen depending on the modulation frequency ω_x, respective the corresponding period T_x, either with a fixed ratio $\nu_x = T_x/2\sigma_x = 1$ to obtain a 2D wavelet prototype or by allowing a certain range $\nu_x = 1..3$ with individual values for T_x being optimized in the automatic feature selection process. The infinite support of the Gaussian envelope is cut off at $1.5\sigma_x$ from the center. For time dependent features, n_0 is set to the current frame, leaving k_0, ω_k and ω_n as free parameters. From the complex results of the filter operation, real-valued features may be obtained by using the real or imaginary part only. In this case, the overall DC bias was removed from the template. The magnitude of the complex output can also be used.

Special cases are temporal filters ($\omega_k = 0$) and spectral filters ($\omega_n = 0$). In these cases, σ_x replaces $\omega_x = 0$ as a free parameter, denoting the extent of the filter, perpendicular to its direction of modulation.

Alternatively, the filter can be designed as the product of a Hanning envelope $h(n,k)$

$$h(n,k) = 0.5 \; - \; 0.5 \cdot \cos\left(\frac{2\pi(n-n_0)}{W_n+1}\right) \cdot \cos\left(\frac{2\pi(k-k_0)}{W_k+1}\right). \tag{3}$$

and the sinusoidal function $s(n,k)$ as above, yielding the window lengths W_n and W_k as parameters instead of σ_n and σ_k (c.f. Fig. 2.1.2 b and d).

2.1.3 Feature set optimization

The main problem of LSTF is the large number of possible parameter combinations. This issue may be solved implicitly by automatic learning in neural networks with a spectrogram input and a long time window of, for example, 1 s. In contrast to this, the time window in the feature extraction process for mel-scaled cepstral coefficients (MFCCs) is much shorter, typically 10 ms. However, the usage of such large time windows is computationally expensive and prone to overfitting, as it requires large amounts of training data, which are often unavailable. By putting further constraints on the spectro-temporal patterns, the number of free parameters can be decreased by several orders of magnitude. This is the case when a specific analytical function, such as the Gabor function (Kleinschmidt, 2002c), is explicitly demanded. This approach narrows the search to a certain sub-set and thereby some important features might be ignored. However, neurophysiological and psychoacoustic knowledge can be exploited for the choice of the prototype, as it is done here.

Feature set optimization is carried out by a modified version of the Feature-finding Neural Network (FFNN). It consists of a linear single-layer perceptron in conjunction with an optimization rule for the feature set (Gramß and Strube, 1990). The linear classifier guarantees fast training, which is necessary because in this method for feature selection the importance of each feature is evaluated by the increase of RMS classification error after its removal from the set. This 'substitution rule' method (Gramß, 1991) requires iterative re-training of the classifier and replacing the least relevant feature prototype in the set with a randomly drawn new one. In the following, an overview of the optimization algorithm is given:

1. Choose M feature prototypes arbitrarily

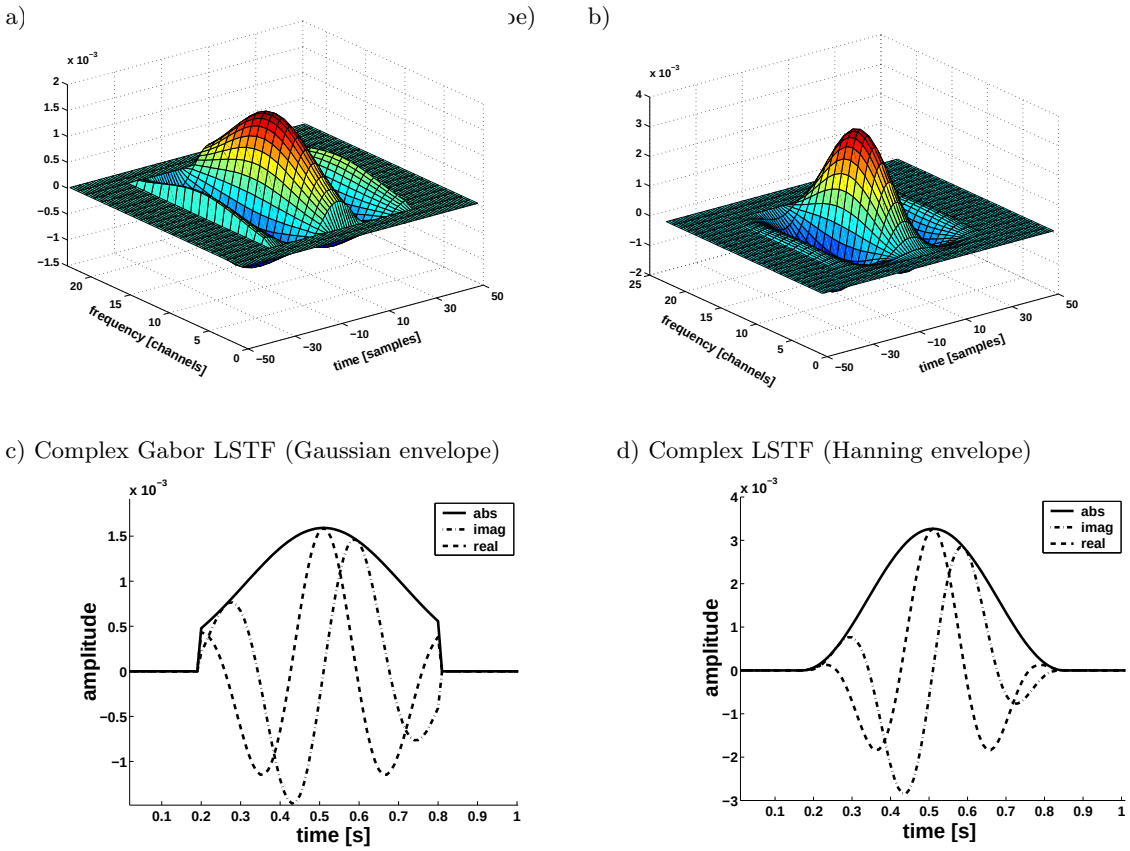

Figure 4: Illustration of 1- and 2-dimensional filter prototypes for LSTFs with Gabor envelope (left panel, support reduced to $[-1.5\sigma \;\; 1.5\sigma]$) and Hanning envelope (right panel). In the top row the real part of complex 2D impulse responses is depicted. The bottom row shows real and imaginary parts as well as envelope of one dimensional LSTFs, corresponding to a cross section of a two dimensional LSTF.

2. Find the optimal weight matrix W using all M feature prototypes and the M weight matrices that are obtained by using only $M-1$ features, thereby leaving out every feature once.

3. Measure the relevance of each prototype by i by

$$R_i = E(without\ prototype\ i) - E(with\ all\ prototypes)$$

4. Discard the least relevant filter $j = argmin(R_i)$ from the subset and randomly select a new candidate.

5. Repeat from 2. until the maximum number of iterations is reached.

6. Recall the set of filter functions, that performed best on the validation set and return it as result of the substitution process (modification of substitution rule).

When the linear network is used for digit classification without frame by frame target labeling, temporal integration of features is carried out by simple summation of the feature vectors over the whole utterance, yielding one feature vector per utterance as required for the linear net. The FFNN approach has been successfully applied to digit recognition in combination with Gabor features in the past (Kleinschmidt, 2002c,a).

2.2 Feature Transformation

One approach to coping with the problem of excessive dimensionality is to reduce the dimensionality by combining feature components, which at the same time reduces computational cost. Linear combinations are particularly attractive because they are simple to compute (e.g. by matrix multiplication) and analytically tractable. Additionally, these transformations are used in ASR to decorrelate the data and thereby enhance the distribution in feature space (Somervuo et al., 2004).

Decorrelated feature vectors are crucial to performance when a Hidden Markov Model with diagonal covariance matrix is used as acoustical classifier. The application of the linear transformations LDA and PCA, that are presented here, can thus help to improve overall accuracy of an ASR system.

Apart from linear transformations, non-linear transformations may be used. These can be implemented with a multi-layer perceptron (MLP), as described in section 2.3.2.

2.2.1 Principal Components Analysis

PCA finds such basis vectors, that represent the data optimal in a sum-squared error sense. It is assumed that the directions with the largest variances are the most important (or the most "principal").

Principal components are determined by calculating the eigenvalues of the covariance matrix associated with the feature vectors and subsequently determination of the eigenvectors. Higher eigenvalues correspond to more important feature vector components (Somervuo et al., 2004). The transformation derived from PCA is the Karhunen-Loéve Transformation (KLT).

Although PCA finds components that are useful for representing data, there is no reason to assume that these components must be useful for discriminating between data in different classes. If we pool all samples, the directions that are discarded by PCA might be exactly the directions that are needed to distinguish between classes. An example for this is shown in Figure 5, where $\overrightarrow{\sigma_1^2}$ contributes to most of the variance, so PCA would identify this vector as the most important principal component. Mapping the data in a one-dimensional subspace using KLT would render the different classes indistinguishable. Where PCA seeks directions that are efficient for *representation*, discriminant analysis seeks directions that are efficient for *discrimination*.

2.2.2 Linear Discriminant Analysis

Linear discriminant analysis (LDA) attempts to find such basis vectors that the linear class separability is maximized. To achieve this, two matrices are computed, the within-class scatter matrix (covariance matrix) S_w and between-class scatter matrix S_b. S_w is a weighted linear sum of class-wise covariance matrices and S_b can be defined as

$$\frac{1}{N}\sum_i n_i(\mu_i - \mu)(\mu_i - \mu)^T$$

where μ_i is the mean of the ith class, n_i the sample count, μ the global mean, N the total number of samples (all classes) and T denotes the transpose.

LDA basis vectors are now the eigenvectors of the matrix $S_w^{-1}S_b$. For c classes, there are at most $c-1$ linearly independent eigenvectors. Not all of them need to be used, but the selection can be based on the eigenvalues, as in PCA (Somervuo et al., 2004).

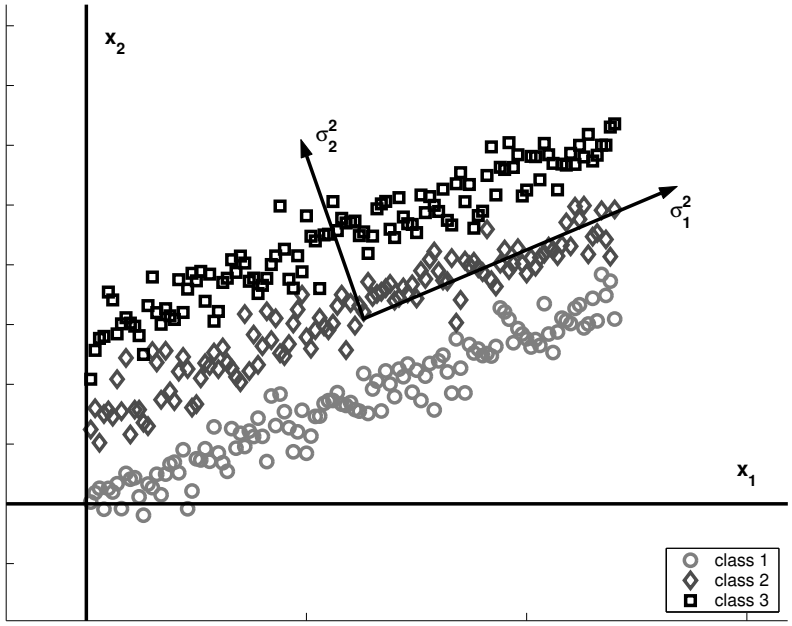

Figure 5: Demonstration of the Adidas problem (which carries its name because of the three stripes that are recognizable in the data set), adapted from (Schukat-Talamazzini, 1995): A PCA would identify $\overrightarrow{\sigma_1^2}$ as principal component, so reducing the data set to one dimension would render the three classes indistinguishable. In order to solve this problem, transformations that employ class information like LDA can be used.

The requirements for the LDA are that each class is modeled by a single Gaussian and the covariance matrices of all classes are equal. Depending on the classes and original features, this can be quite far from true distributions, so non-linear feature transformations might be necessary, where these limitations do not apply. A multi-layer perceptron is tool too achieve such a transformation, as in the Tandem approach presented in section 2.3.2.

2.3 Classifiers - HMMs and the Tandem Approach

The problem of classification is to find the correct transcription given a sequence of feature vectors and can—in a statistical sense—be defined as follows:

Let $\mathcal{X}$ be the set containing all possible feature vectors and $\mathcal{V}$ the vocabulary of the classifier. Given a sequence of feature vectors

$$X = x_1, x_2, ..., x_m \text{ where } x_i \in \mathcal{X}$$

what is the most probable word sequence

$$W = w_1, w_2, ..., w_n \text{ where } w_i \in \mathcal{V} \text{ ?}$$

To solve this problem, one can search for the word sequence that maximizes the term

$$\hat{W} = \underset{W \in \mathcal{V}}{argmax}\ P(W|X)$$

The *a-posteriori* probability $P(W|X)$ is not directly accessible, so the problem is reformulated using Bayes rule:

$$P(W|X) = \frac{P(W) \cdot P(X|W)}{P(X)}$$

where $P(W)$ is the probability for the occurrence of W and $P(X|W)$ the likelihood of X, given the word sequence W. P(X) (the probability of the occurrence of X) is independent of W and can thus be ignored in the following considerations.

The probabilities $P(W)$ are a statistical measure for plausibility of syntax and semantic of W and can be calculated by using a language model. According to Bayes rule, $P(W)$ can be calculated by

$$P(W) = \prod_{i=1}^{n} P(w_i|w_1, ..., w_{i-1}) \tag{4}$$

In order to keep the complexity of this model at a reasonable level, it is commonly assumed that the probability for a word depends only from the previous two (trigram language model). This yields the approximation

$$P(W) \approx \prod_{i=1}^{n} P(w_i|w_{i-2}, w_{i-1}) \tag{5}$$

So the language model stores the probability of occurrence for each combination of a sequence of three words. Optimally, these probabilities reflect the application type of the ASR system. A language model is only necessary for corpora, where sentences with semantic meaning are processed. The usage of a language model for digit recognition systems is not expedient in most cases.

$P(X|W)$ can be maximized with an acoustic model, for which HMMs and ANNs are the most commonly used methods.

2.3.1 Hidden Markov Models (HMMs)

The HMM approach is a well-known and widely used statistical method of characterizing the spectral properties of the frames of a pattern. In ASR systems, HMMs are commonly used to calculate the likelihoods of phonemes, words and sentences. In the following, Markov Chains and an extension to these, the Hidden Markov Models, are introduced.

Consider a system that may be described at any time as being in one of a set on N distinct states indexed by $1, 2, ..., N$. At regularly spaced, discrete times, the system undergoes a change of state (possibly back to the same state) according to a set of probabilities associated with the state. We denote the time instants associated with state changes as $t = 1, 2, ...$ and we denote the actual state at time t as q_t. This model is called a first-order Markov chain if the current state q_t depends solely of the previous state q_{t-1} and the transition probabilities $a_{ij} = P(q_t|q_{t-1})$ can be combined in the matrix $A = [a_{ij}]_{N\times N}$ with $\sum_j a_{ij} = 1$ and $a_{ij} \geq 0 \; \forall \; i, j$. The probabilities of the initial state are given by the vector $\overrightarrow{\pi} = P(q_1 = s_i)$.

A stochastic process is called Hidden Markov Model (HMM), if the following, additional requirements are met: A second process for each point in time t emits an element of a finite output set $\mathcal{K} = \{\nu_1, \nu_2, ..., \nu_K\}$ in dependency of the current state q_t and only the output sequence $\mathbf{O} = (O_1, O_2, ..., O_T)$ is known to the observer, whereas the state sequence remains hidden.

In the discrete case, the output distribution can be described by

$$\mathbf{B} = [b_{jk}]_{N\times K} \text{ with } b_{jk} = b_j(\nu_k) = P(O_t = \nu_k|q_t = s_j) \tag{6}$$

where $\sum_k b_{jk}$ and $b_{jk} > 0 \; \forall \; j, k$.

In the case of continuous output distributions $B_j(\vec{x})$ with $\vec{x} \in R^D$ for a D-dimensional output space usually multivariate normal distributions are employed. Typically, a mixture of Gaussian distributions is used to model output distributions. HMMs with such distributions are called Gaussian mixture models (GMMs).

The dependencies between the components of $\vec{x}$ is given by the covariance matrix. In case of completely independent feature vector components, a diagonal covariance matrix is used. For each state q_t either a superposition of different distributions can be considered (mixed densities) or - in case of semi-continuous Markov models - the same distribution for all states can be used.

The Markov model is completely determined by the parameter set

$$\lambda = (\overrightarrow{\pi}, \mathbf{A}, \mathbf{B}) \tag{7}$$

and the number of states N as well as the extent of the output set K.

Three problems arise, when HMMs are applied to solve the problem of ASR:

- How can the likelihood $P(O|\lambda)$ for a given set of parameters λ be calculated?
- For a given model λ, how can the most probable state sequence $\mathbf{q}$ be determined for an observed emission sequence O?
- How can the set of parameters $\lambda = (\overrightarrow{\pi}, \mathbf{A}, \mathbf{B})$ be optimized, such that the distribution $P(O|\lambda)$ optimally corresponds to the events that are to be modeled?

The straightforward solution to the first problem is to enumerate every possible state sequence of length T (the number of observations). As there are N^T such sequences, computational cost is extremely high, so this is not a feasible method. Similarly, for the other two problems computational cost for the trivial solution is much too high to be practically applicable.

Luckily, for each of these problems there exist a number of solutions that are not as computational expensive, the most prominent being the forward-backward procedure, the Viterbi algorithm and the Baum-Welch algorithm. A detailed description of these can be found in (Rabiner and Juang, 1993).

Spoken language does not meet the pre-requisites of a Markov process, as the current state usually depends on more than one of the previous states. Nevertheless, HMMs are successfully applied to problems in ASR.

Since feature vector components can reach any value, the acoustic model is commonly obtained with a GMM. From the calculated likelihoods a conventional HMM with discrete observation distributions is employed, in order to determine the most probable word sequence. Both models may be combined in one structure, as it was the case in all experiments presented in this work.

2.3.2 The Tandem Approach

The Tandem approach to ASR is based on a conventional GMM-HMM recognizer combined with an artificial neural network (ANN) as additional acoustical classifier.

The setup is build up as follows: A non-linear multi-layer perceptron (MLP) with one hidden layer uses a sequence of feature vectors as input to calculate subword (phones or diphones) posterior probabilities. The network is trained by backpropagation to targets obtained from hand labeling or forced-alignment, for which usually word-level transcripts

of the utterances are used and the word sequence is used to constrain an optimal alignment between existing speech models and the new speech data. The network output is transformed and used as input for a conventionally trained GMM-HMM model. Because of the skewed distribution of MLP output values, either the logarithm of these values is calculated or the final non-linearity of the MLP is left out. The non-linearity typically used is softmax, which is used because the outputs of a ANN should be interpretable as posterior probabilities for a categorical target variable, so the outputs should lie between one and zero and sum to one. The purpose of the softmax activation function is to enforce these constraints on the outputs. Let the net input to each output unit be $q_i, i = 1, ..., c$, where c is the number of categories (or output neurons). Then the softmax output p_i is:

$$p_i = \frac{\exp(q_i)}{\sum_{j=1}^{c} \exp(q_j)} \tag{8}$$

From another point of view, the MLP can be seen as part of the feature extraction stage, as it applies a transformation to feature space very similar to LDA, where the classes that are separated are phones or diphones. A difference to LDA is that the transformation is non-linear.

A reason for the good performance that can be achieved with this approach might be that neural networks focus their modeling power to the regions in feature space, were large variability is present and which are therefore difficult to model by the HMM. By transforming the feature space, these regions are enlarged, while others, not as important regions, are only coarsely mapped to the new feature space, so the modeling task is simplified.

For our experiments, the tandem approach was chosen as it has proved to give superior performance compared to GMM-HMM systems Hermansky et al. (2000).

3 Corpora

Corpora are collections of speech material, that are used for ASR systems in order to provide training- and test data for the statistical models, as described in the previous section. A universal translator as in the Star Trek universe, that recognizes connected word sequences in *any* language, can handle large vocabularies and shows perfect recognition performance in acoustic adverse conditions, is not (yet) existent [2].

Thus, decisions regarding the research objectives or the type of application have to be made when designing ASR systems: If a close-talk microphone is utilized for recording speech like in dictation software, a large vocabulary plays a more important role than robustness. For information services by telephone on the other hand, invariance of performance in the presence of channel disturbances or independency from speaker identity or gender are important properties. Furthermore, today's ASR systems are limited to recognition of one language, which is therefore another important design parameter.

These decisions retroact on the choice of training- and test material, so that different corpora are needed to investigate the various questions posed in this work. These corpora are presented in the following.

3.1 TIDigits

The TIDigits corpus contains speech which was collected for the purpose of designing and evaluating algorithms for speaker-independent recognition of connected digit sequences. There are 326 US-American speakers (111 men, 114 women, 50 boys and 51 girls) each pronouncing 77 digit sequences. Each sentence contains up to 7 digits, which are mainly monosyllabic words. The corpus was collected at Texas Instruments (TI) in a quiet acoustic enclosure with a sampling rate of 20 kHz.

3.2 Aurora 2 corpus

The corpus is part of the Aurora 2 framework (Hirsch and Pearce, 2000), that has been developed by Ericsson Eurolab Germany and Motorola Labs for evaluation of feature extraction methods.

For the Aurora 2 corpus, clean speech material from the TIDigits database (adult speakers only) resampled to 8 kHz was mixed with eight different noise signals at specific signal-to-noise ratios (SNRs), ranging from -5 dB to 20 dB in 5 dB steps. The speech material was divided into one training and three test sets.

Two training modes were defined, using either clean speech data only or multi-condition data (i.e. data that contains both clean and noisy signals). The multi-condition training set contained speech mixed with four noise signals, namely suburban train, crowd of people (babble), car and exhibition hall.

Testing is carried out with multi-condition data with seven different noise conditions (clean plus the earlier mentioned noise signals at six different SNRs). The first test set is test A, where the same noise signals as for the training corpus have been used (matched

[2] In order to construct the Star Trek universal translator, apart from a perfect ASR system, some other inventions like automatic speech-understanding, translation and lip-synchronous play back of synthesized speech are missing. Additionally, the universal translator is capable to produce complete, error-free dictionaries with only a few sentences of training material - which is a quite an ambitious research objective.

training-test-condition). Test B contains speech mixed with the four remaining noises (restaurant, street, airport and train station). For test C, speech signals are filtered with a telephone bandpass characteristic before applying the noises suburban train and street. The testing procedure yields word error rates in dependency of SNR, test set and noise signal. See table 21 as an example.

The Aurora 2 paradigm aims specifically at robust feature extraction techniques, and is therefore very well suited to the scope of this thesis. In order to evaluate robustness of a system, results for the clean trained HMM are especially interesting, as the HMM models do not contain any information about possible distortions in this case. Test B for the multi-condition setup and test C for both training modes are also of interest in this context because of the mismatch of noise signals in training and test or the mismatch of frequency characteristics.

3.3 TIMIT

TIMIT is a phoneme-labeled corpus that contains 6300 phonetically balanced sentences with continuous speech from a total of 630 speakers, coming from 8 different dialect regions in the USA. Like TIDigits, it was recorded at TI; most of the labeling was carried out at the Massachusetts Institute of Technology (MIT). In contrast to the other corpora described here, where male and female speakers are equally represented, the percentage of female speakers is only 30 %.

The original TIMIT corpus was recorded in an acoustically clean environment. In our experiments, it was used as training database for the neural net of the Tandem recognition system, that calculates the likelihood of the occurrence of a phoneme, given a feature vector sequence (see subsection 2.3.2). Hence, the training corpus has to be phoneme-labeled. Many of the experiments presented in this work were carried out using the Aurora 2 corpus. To account for this, the TIMIT speech signals were mixed with the noise signals present in Aurora 2, in order to improve overall performance. However, statistics for PCA were computed with the clean speech TIMIT data.

3.4 Zifkom

The Zifkom database was created by the German Telecom and consists of 2000 sentences spoken by 100 female and 100 male speakers, where each sentence contains one German digit or command word.
The corpus was used for feature set optimization, where only the sentences containing digits were employed, so target words were mainly mono-syllabic. It was equally split into a training and a test set; for feature selection on noisy data, Aurora noises were added to the speech files with the SNRs defined by Aurora 2 as for the TIMIT data. For the selection of feature prototype sets, either the clean or noisy Zifkom database was used.

3.5 CarDigit

This corpus is used at Philips Research Labs (see section 5) and was used to evaluate performance of LSTF features in combination with real-world recordings. It contains word-labeled German digit strings recorded at 16 kHz sampling rate in automotive and office environments and consists of several subsets, one of which is SpeechDatCar (that originates from the homonymous project). CSDC is another subset and was produced

within the framework of the German project MoTiV ("Mobilitaet und Transport im intermodalen Verkehr").

CSDC and SpeechDatCar, as well as the subset "office" (containing close-talk data, recorded in office environment) were used as training data. The several test subsets originate from miscellaneous projects or internal tests from Philips and contain car-recordings only.

As the mean SNR of the test set is about 10 dB with a rather static background noise, the recognition task can be considered as easier as for the Aurora 2 test sets. Altogether, the corpus builds up a broad and realistic distribution of car environments for German language.

3.6 CarCity

Like CarDigit, CarCity is a heterogenous speech base used at Philips Research Laboratories. As the name suggests, training- and test data of this corpus consist of city names. Due to the large vocabulary (2935 or 10139 city names, depending on the test set), phoneme-based ASR systems are used in combination with with corpus instead of whole-word models that are commonly used for digit recognition tasks.

Just as for the CarDigit corpus, the acoustic data used to train and test the systems are real world in-car recordings (sampled at 16 kHz) which reflect the true automotive environmental conditions. The database covers two languages: German and English, where for our experiments only German language was used.

The speech material used to train the German phoneme models is a collection of two different sub-corpora: CityTrain and sdc (SpeechDatCar). The CityTrain and sdc data sets are recorded with a far-field microphone (CityTrain) or with both a far-field and a close-talk microphone (sdc). As the test corpora contained only far-field recordings, the sdc close-talk data was not used for training. The training corpus covers 8 cars and 850 speakers.

The test corpus is named CityTest, for which close-talk and far-field recordings were available. The far-field recordings used in our experiments have an average SNR of 10.1 dB. The test sets with 2935 and 10139 city names will be referred to as CityTest-3k and CityTest-3k , respectively.

4 Evaluation and Improvement of Localized, Spectro-Temporal Filters

In this section, two questions regarding features localized, spectro-temporal filters (LSTFs) as proposed by Kleinschmidt (2002b) are investigated:

Firstly, what are the optimal parameters of features derived from LSTFs? This question is posed because one of the downsides of LSTF features is the large number of vector components (compared to standard feature extraction methods) in previously presented experiments, which is accompanied by high computational load. The high dimensionality arises from the relatively high number of filter prototypes in each set and the concatenation with delta and double-delta dynamic features. To answer this question, the performance of LSTF features is determined with different numbers of filter prototypes (section 4.3). Furthermore, the necessity of dynamic features is investigated in section 4.3.

The second question is: By what means can the robustness and the overall performance with LSTF features be increased? As our work is led by the biological blueprint, physiological constraints are considered in order to achieve this. Additionally, knowledge from signal-processing will also be employed. Therefore, the cut-off Gaussian envelope in the LSTF function will be replaced with an Hanning envelope, in order to determine whether the improved modulation-frequency characteristics affect robustness and recognition performance (section 4.5.

The spectro-temporal receptive field (STRF) has properties, that are not fully exploited in the original Gabor approach: STRF patterns usually exhibit only one maximum and the STRF transfer function is separable. It was investigated, whether changes to the modulation filters that account for these findings help to improve robustness. The experiments, where the number of maxima of LSTFs was limited to one are described in section 4.6. The usage of separable filter functions is investigated in section 4.7.

The method to evaluate and improve the feature sets is given by the Aurora 2 framework, where noisy digit strings (as described in the previous section) are used to train and test a Hidden Markov model. The properties of the framework are described in section 4.1. A description of the recognition system, which contains a non-linear artificial, neural network as suggested in the Tandem approach, is presented in section 4.2.

A number of different modulation filter prototype sets were calculated in our experiments. An overview these is given in table 1.

feature set	training corpus	description	best set
G1	TIMIT	These sets were proposed in (Kleinschmidt, 2002) and evaluated in subsection 4.3 and 4.4.	-
G3	zifkom		-
HBxx	zifkom	Hanning envelope (c.f. section 4.5)	HB02
GBxx	zifkom	Gaussian envelope, sets were generated as comparison to HBxx (c.f. section 4.5)	GB03 / GB07
HEWxx	zifkom	Hanning envelope, number of oscillations $\mathbf{v}_x$ limited to one (c.f. section 4.6)	HEW04
SEPxx	zifkom	Hanning envelope, fully separable filter functions (c.f. section 4.7)	SEP06

Table 1: Overview of different feature prototype sets. The differences to the reference filter sets, as well as the best set in a list of prototype sets are presented.

4.1 Experimental Framework: Aurora 2

Experiments with the Hidden Markov Toolkit (HTK) setup were carried out in the Aurora2 framework (Hirsch and Pearce, 2000): The Aurora 2 speech corpus, as described in section 3, was used to evaluate existing and new feature sets, where the task is to recognize clean and noisy digit strings. Aurora 2 baseline results are obtained with 12 MFCCs with deltas and double deltas, yielding a feature vector dimension of 39. No noise suppression was applied to the speech data before computing the cepstral coefficients.

The results given are either absolute word error rate (WER) or relative reduction of WER. The WER is the sum of insertions, deletions and misses, divided by the total number of words. Averaged WERs for Aurora are calculated by averaging over all test subsets and the SNRs 0, 5, 10, 15 and 20 dB. Results for clean test and -5 dB SNR are not included in the averaged results.
Commonly, relative reduction in word error rate R_{WER} (or the relative improvement) is also reported besides absolute values. For Aurora 2, it is calculated by

$$R_{WER} = \frac{1}{N \cdot M} \sum_{n=1}^{N} \sum_{m=1}^{M} \frac{WER(n,m)_{Base} - WER(n,m)_{Exp}}{WER(n,m)_{Base}} \tag{9}$$

where $WER(n,m)_{Base}$ is the Aurora baseline result and $WER(n,m)_{Exp}$ is the measured result in dependency from the SNR n and the noise type k. N and M are the total number of SNR conditions and noise types, respectively. By this definition, differences between WER_{Base} and baseline are more emphasized the better the baseline result is. This is reasonable, as a constant performance gain is more valuable for a system with already low WERs.
Another important factor when evaluating ASR systems is the sentence error rate (SER), which is the number of incorrectly recognized sentences divided by the total number of sentences. A sentence is regarded as erroneous, if it contains an incorrectly identified word, e.g. if an insertion, deletion or substitution occurs.

4.2 Experimental setup

From the Aurora 2 corpus and a set of LSTF prototypes, secondary features were computed according to section 2.1 and fed into a tandem recognition system as described in section 2.3.2. The feature vectors with 60 to 80 components are online normalized (yielding features with zero mean and variance of 1) and combined with delta and double-delta derivatives. They are subsequently fed into the multi layer perceptron (MLP) with 60 or 80 input neurons, 56 output neurons and 1000 neurons in the hidden layer [3]. The MLP was trained on the TIMIT phone-labeled database by backpropagation with artificially added noise, as described in section 3. Because of the skewed distribution of MLP output values, the softmax non-linearity (see equation 8) was left out.

The 56 output values were then decorrelated via PCA (statistics derived on clean TIMIT) and fed into a fixed HTK [4] back end, which was configured according to the Aurora 2 experimental framework.

In this setup, both a Gaussian mixture HMM (GMM) and a conventional HMM are combined in one lattice structure, that represents both acoustical and word model. Because we followed the the Aurora 2 paradigm, the GMM-HMM system uses a relatively

[3] QuickNet software package provided by ICSI, http://www.icsi.berkeley.edu
[4] Software used was HTK V2.2 from Entropic

small number of parameters, which lowers computational cost but also decreases overall performance. The system is thus referred to as 'small-footprint system'. It was trained on Aurora 2 multicondition or clean only training data as explained in section 3.

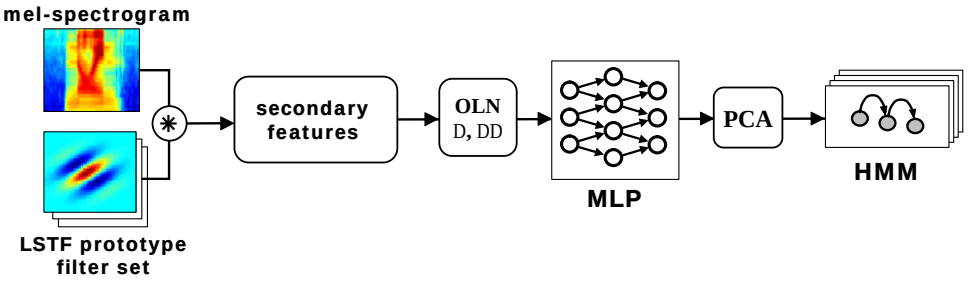

Figure 6: Schematic overview of the experimental setup. Feature vectors are obtained from correlation of mel-spectrograms with LSTF prototypes and fed into a Tandem recognition system. See text for further description.

In the first two experiments (4.4 and 4.6) features were computed using the sets G1 and G3 from (Kleinschmidt and Gelbart, 2002) which were optimized on noisy, American English conversational speech or noisy German digits (ZIFKOM corpus), respectively. G3 yields relative improvements of over 50 % compared to the baseline for clean training in a single stream experiment and improvements of 36 % and 74 % for noisy and clean training, respectively, in a multi-stream combination with the Qualcomm-ICSI-OGI front end (Adami et al., 2002). The results presented in the following are averaged word error rates or relative improvements obtained with the Aurora 2 test corpus, which contains a total of 50050 sentences with 164415 words. Calculation of averages was carried out according to section 4.1.

4.3 Necessity of delta and double delta derivatives

Deltas and double deltas (also known as dynamic features) can be regarded as numerical approximations to local first and second order derivatives, respectively, and correspond to FIR lowpass and bandpass filters. They are calculated by convolving the feature vector components with a 9-point impulse response $h(i)$ in order to emphasize speech components with a relatively high rate of change

$$\Delta c(n) = \frac{\sum_{i=1}^{k} h(i)c(n+i)}{2k+1}$$

where a linear impulse response $h(n)$ is used to derive deltas and a parabolic $h(n)$ to compute double deltas.

Thus, they are used to account for information inherent to temporal dynamics. As cepstral coefficients neglect temporal information, usage of deltas greatly increases performance for this feature type and is therefore commonly used in today's ASR systems. Tests with the recognition system `ASPIRIN` (described in section 5) showed, that WER on Aurora 2 is improved by 30 to 35 % relative when the deltas are used. Adding double deltas to the feature stream usually results in an additional gain of 6 % relative improvement (Ellis and Gelbart, 2004).

For experiments in (Kleinschmidt, 2002b) deltas were also used in conjunction with Gabor features. As Gabor features incorporate temporal, spectral and spectro-temporal

information, it was investigated, whether the usage of temporal derivatives is beneficial. This is an important parameter in the evaluation process: If dynamic features do not contribute to overall performance, these can be left out, so feature vector dimensionality would be reduced by 67 % with the effect of heavily reduced computational cost.

Recognition results were obtained with the HTK system as described in 4.2 using the feature prototype sets G1 and G3 with 60 feature components. A detailed description of these sets can be found in (Kleinschmidt, 2002b). Performance was determined for different setups, where either deltas and double deltas, only first-order derivatives or no deltas at all were used, yielding feature vector dimensions of 180, 120 or 60, respectively. Fewer feature vector components result in less input neurons for the neural network and thus in decreased number of weights. To keep the complexity of the acoustic classifier constant, the number of hidden neurons was adjusted, so that the total number of weights was the same for all three tests.

Results for G3 and G1 are almost identical, so only WERs for G1 are reported here. Absolute word error rates in dependency of the SNR are shown in Figure 7. While improvements can be achieved by using delta features, performance gain is not as dramatic as for cepstral coefficients as described above. The benefit for first order derivatives in terms of absolute WER ranges from 0.5 for clean condition test to 5.8 % for a SNR of 0 dB. The averaged relative improvement is 6.3 %.
By adding second order derivatives only slight improvements can be obtained, so the lines denoting D1 and D2 in Figure 7 are difficult to distinguish. Double deltas give at most another 1.1 % better absolute WER. For high SNRs (15 dB and better) improvements are much smaller, ranging between 0.1 and 0.3 percent. Averaged, relative improvement is 1.7 %. These differences are are very small compared to the MFCC results.

Dynamic features increase the performance for ASR systems with LSTF features, albeit not as dramatic as for systems with cepstral coefficients. First-order derivatives improve results especially in low SNRs and therefore contribute to robustness. Adding double deltas brings merely slight enhancements. For systems were the last bit of performance is not as important as computational time, these can be omitted, decreasing the feature vector dimensionality by 33 %.
The reasons for the general reduced error rates with deltas are the properties of the filter sets. Apart from purely temporal and spectro-temporal modulation filters, the sets G3 and G1 also contain purely spectral filters, so the data added by deltas leads to a gain in information. For the set G1, the fraction of purely temporal modulation filters is 38 %, for G3 it is 30 %.

4.4 Optimal number of features

Higher number of features require more computation time and do not necessarily lead to improved recognition performance. It is therefore desirable to determine the optimal number of LSTFs. In this experiment the number of features used as input for the tandem system was varied from 10 to 80 features. A reduction of number of features would result in fewer input neurons for the MLP, thus decreasing the total number of weights. As for the delta-experiment, the number of neurons in the hidden layer was adjusted for a fair comparison of classification performance, so that the total number of weights remained constant at about 180,000.

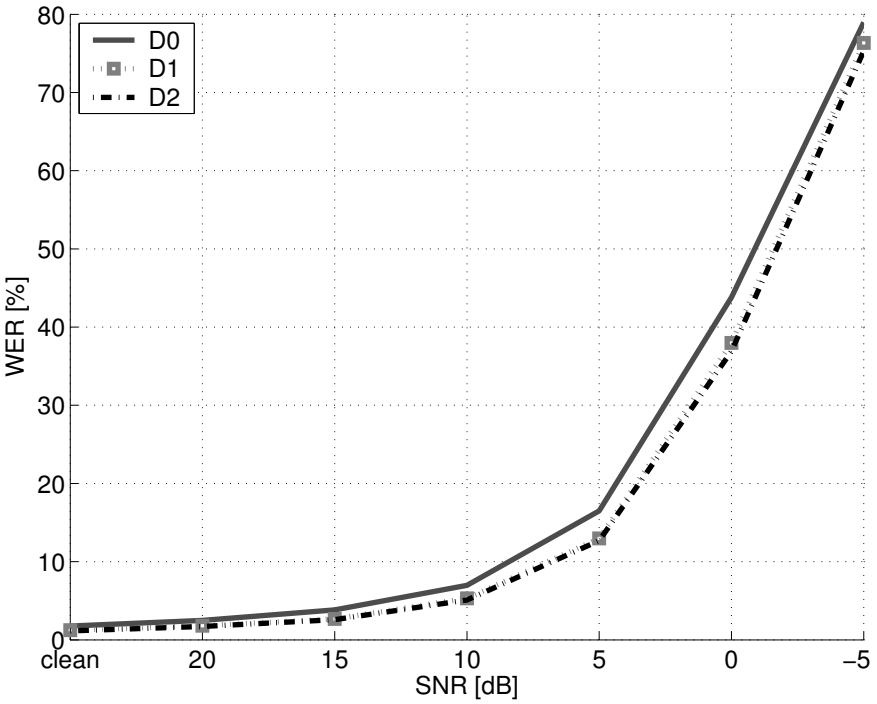

Figure 7: Word error rate in dependence from SNR for feature streams without derivatives (solid line), with first-order deltas (dotted with marker) and with deltas and double-deltas (dash-dotted line)

The feature set G3, which was used in this experiment, consists of 80 feature prototypes ordered by relevance. When using less than 80 features, the most relevant prototypes were chosen.

In Fig. 8 the obtained error rates are shown. While WERs for multi condition training steadily increase with higher number of features, this is not the case for clean condition training, where performance drops when using 80 instead of 70 features. However, both curves show saturation at 60 features, while performance superior to the baseline results is already achieved with 50 features for multi-condition training and 20 features for clean-condition training. The optimal number of features in the set would depend on application restrictions. Acceptable performance is reached with as few as 30 and optimal performance with 70 features for set G3. The increase in WER from 70 to 80 features indicates that the least important 10 features in the set even have a detrimental effect on recognition performance, possibly a result of the optimization algorithm (c.f. Section 2.1.3).

4.5 Envelope optimization

Cutting off the support of the Gaussian envelope at 1.5 σ as shown in Figure 2.1.2 results in unwanted higher harmonic frequencies in the modulation frequency domain. These distortions can be eliminated to a great extent by replacing the Gaussian envelope with a Hanning window. Fig. 4.5 shows a comparison of the spectro-temporal modulation transfer function of the two filter types.

In order to determine if the favorable modulation frequency characteristics of Hanning envelopes lead to improved recognition performance, several prototype sets were calculated. The training process is not deterministic, because the filter functions are randomly

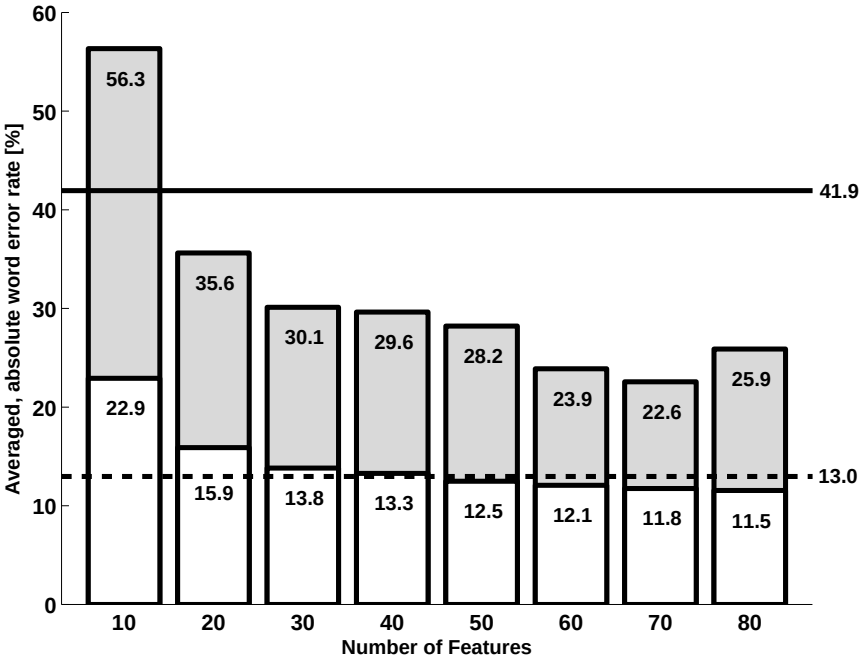

Figure 8: Averaged recognition performance for different number of features: results are shown for clean condition training (grey) and multi condition training (white). Baseline results are plotted as horizontal lines for multi condition training (dashed) and clean condition training (solid).

a) LSTF with cut-off Gaussian envelope

b) LSTF with Hanning envelope

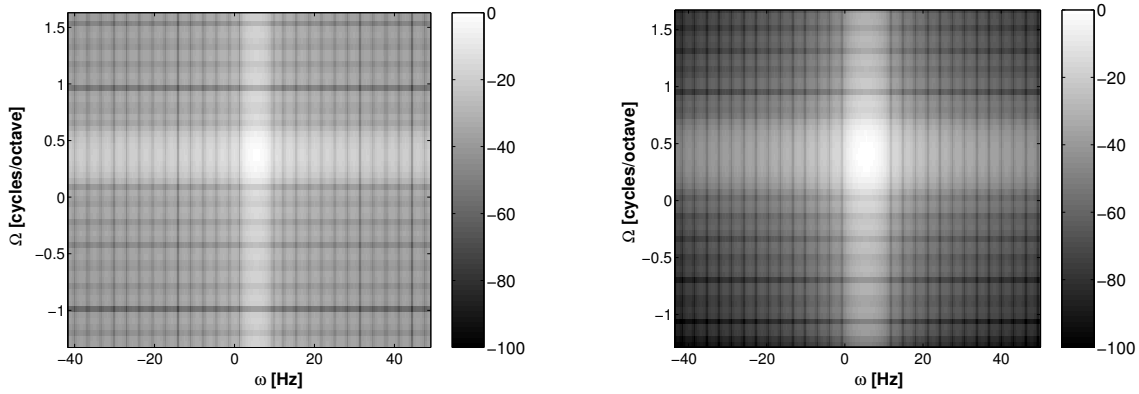

Figure 9: Absolute values of spectro-temporal transfer functions for real part of LSTF prototypes plotted on logarithmic scale. The shading denotes the amplitude in dB.

chosen, so that training with the same parameters yields different prototype sets. To receive more reliable results, eight feature sets with Gaussian and eight feature sets with Hanning envelope were generated by the automatic optimization procedure (Section 2.1.3) with ZIFKOM German digit data. Temporal and spectral modulation frequencies were randomly chosen in an interval from 2 to 50 Hz and 0.06 to 0.5 cycles/octave, respectively. The width of the envelope was loosely coupled to the modulation frequency ω_x, using a value from 1 to 3 for the number of periods ν_x that lie in the interval $[-\sigma_x \;\; \sigma_x]$ for Gaussian envelopes or in the interval $[-W_x/1.5 \;\; W_x/1.5]$ for Hanning envelopes. Boundary conditions for ν_x guaranteed that even at low modulation frequencies

the extension of the prototypes did not exceed 23 frequency channels or 101 time frames (corresponding to 1 second filter length).

Either absolute, imaginary or real part of the filter output were used as features. German digits (ZIFKOM) mixed with different noise conditions were used for optimization. Each set contained 80 feature prototypes, from which the most relevant 60 were used in the experiment.

	average absolute WER		relative improvement	
	multi	clean	multi	clean
a) baseline	13.00	41.90	0.00	0.00
b) G3	12.10	23.90	2.90	50.05
c) Avg Hanning	12.29 ± 0.14	21.57 ± 0.7	1.14 ± 2.27	53.53 ± 1.08
d) Avg Gauss	13.22 ± 0.20	23.67 ± 1.07	-3.09 ± 2.36	49.97 ± 2.33
e) Hanning HB02	12.00	19.49	7.93	58.83
f) Gauss GB07	12.60	23.90	2.55	49.62
g) Gauss GB03	13.10	19.60	-0.15	56.70

Table 2: Word error rates and relative reduction of error compared to the baseline for different feature types. Beside the baseline data (a), results are shown for feature set G3 (b), averaged values with standard deviation for eight Hanning and eight Gaussian envelope sets (c & d) and best Hanning and Gaussian envelope sets (e) - (g)

Beside the Aurora 2 baseline, results are reported for G3 and the averaged error rates for the new generated Hanning- and Gaussian-envelope prototype sets in Table 2. Furthermore, results for the best prototype sets are shown. For Gaussian sets, GB07 showed best performance for multi condition training and GB03 for clean condition training. In the case of Hanning-sets, the set HB02 produces best results in both training conditions. The results show that in average Hanning-shaped LSTFs outperform Gabor-shaped features in all conditions. The best feature set with Hanning envelope HB02 also outperforms the reference feature set G3 and the best LSTF set with Gaussian envelope.
Statistical information regarding the distribution of ν_x and the relative frequency of spectro-temporal features was determined for all feature prototype sets with Hanning envelope. In Figure 10 a), the relative frequency of purely temporal and spectral and spectro-temporal filters are compared. The large percentage of spectro-temporal features indicates the importance of filters, that are able to detect diagonal structures in primary feature matrices.
An optimization problem arises with the width of the prototype envelope relative to the modulation frequency period: The wider the envelope (larger ν_x) the more selective is the filter in modulation frequency domain. However, this benefit comes with the expense of larger prototypes, that contain more complex spectro-temporal patterns, have higher computational demand, and are not very well corresponding to physiological STRFs. In past experiments, 1.5 oscillation periods per feature ($\nu_x = 1$) were chosen ad hoc as a fixed ratio for all features in the set. Allowing for automatic selection of ν_x yields a distribution that peaks close to one as shown in Figure 10 b). Note that no values of $0 < \nu_k < 1$ appear, because the mininum value for ν_k was limited to one and purely temporal modulation filters (to which no ν_k can be assigned) accumulate in the histogramm bin corresponding to $\nu_k = 0$.

This supports the ad hoc defined prototype. However, the overall results support a loose constraint on envelope width, i.e. allowing a certain range might be beneficial since each individual feature may have a slightly different optimal ν_x value.

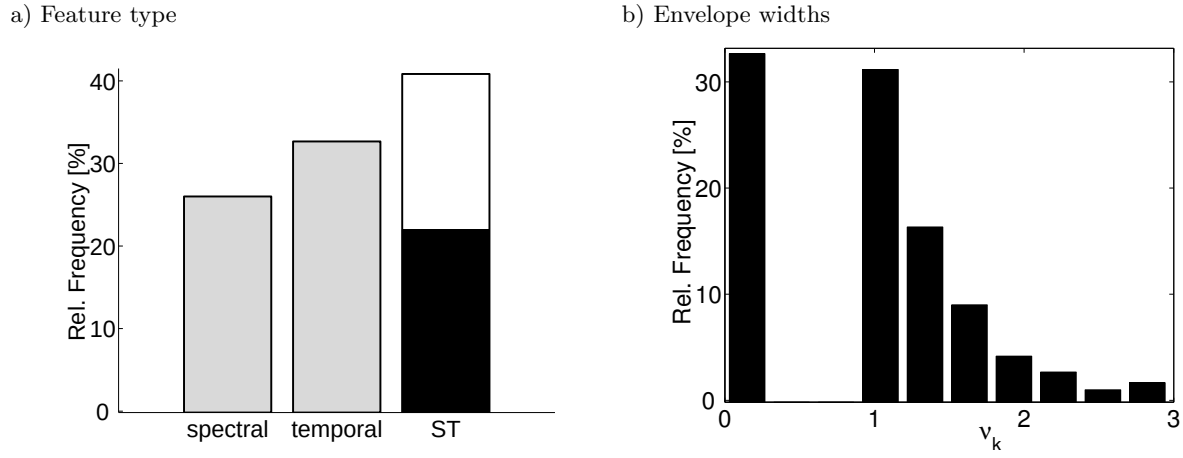

Figure 10: Statistics for feature prototypes with Hanning envelope (total of 640 features). a: Distribution of purely spectral or temporal LSTFs (grey) and spectro-temporal filters. The latter are split in upwards (black) and downwards (white) direction, corresponding to positive or negative temporal modulation frequencies. b: Distribution of the ratio $\nu_k = T_k/2\sigma_k$.

4.6 Comparison of envelope widths

The LSTF prototypes in set G3 show more than one maximum because the interval $[-\sigma_x \quad \sigma_x]$ was chosen to contain exactly one period ($\nu_x = 1$). Still, the support was cut off at 1.5σ, leading to secondary maxima. However, in neurophysiological STRFs commonly only one maximum is observed.
In order to investigate the influence of envelope width, a new feature sets was produced by modifying the existing feature set G3: Halving the values for σ_n and σ_k yields feature set G3sn, where the number of maxima within the Gaussian envelope is limited to one.

Furthermore, seven new prototype sets with the same properties as the modified set G3sn were generated. To obtain these sets, the FFNN selection rules were changed, so that only filter functions with the desired attributes were selected. A Hanning envelope instead of a Gaussian envelope was used, as this proved to give better overall performance. An example for such a feature set is presented in Figure 11. Using the new sets and G3sn, secondary features were computed and fed into the tandem system as described in section 4.2.

Error rates for both the modified set as well as the newly generated sets are shown in table 3. Among the averaged results, where error rates from 0 to 20 dB SNR are included, recognition rates for "high SNR" test conditions are also reported. These are calculated by averaging the values for clean, 15 dB and 20 dB SNR test. Additionally, WERs for the best set with Hanning envelope and changed envelope width (called HEW04) are shown.

While performance could not be increased in general by this physiological motivated modification, error rates can be lowered in high SNR conditions:
The modified set G3sn performs worse than the original G3 for the full tests, while it yields improved results for clean and high SNR test conditions—for the high SNR test, differences in relative improvement are 19.5 % for clean condition training and 2.3% for multi condition training.
The newly generated sets compete with HB02, and results are very similar to the comparison between G3 and G3sn: Overall performance is worse for the new sets (although the best set HEW04 comes very close to HB02), but lower error rates are observed for high SNRs. In this condition, usage of HEW04 reduces the WER compared to HB02

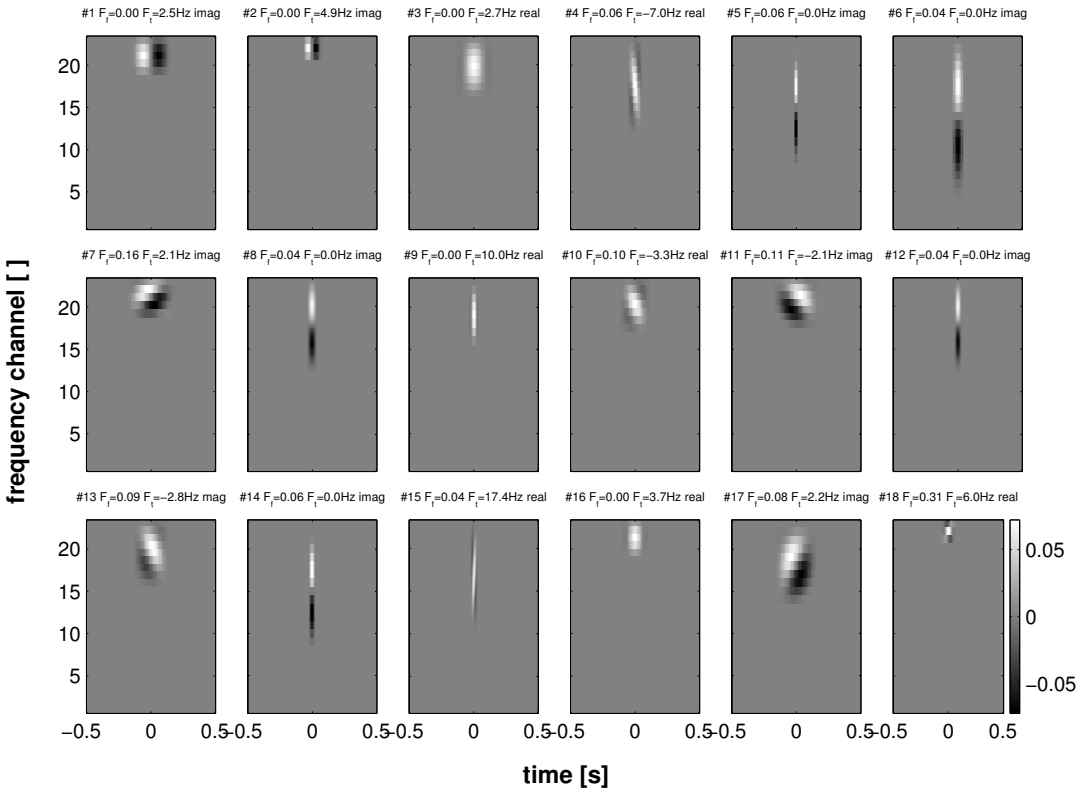

Figure 11: Modulation filters from feature prototype set HEW04. To obtain this set, FFNN parameters were chosen such that filters contain exactly one maximum (and eventually a minimum).

	absolute WER				relative improvement			
Training condition	**multi**		**clean**		**multi**		**clean**	
Test condition	**full**	**high SNR**	**full**	**high SNR**	**full**	**high SNR**	**full**	**high SNR**
Baseline	12.97	2.30	41.94	7.90	0.00	0.00	0.00	0.00
G3	12.08	2.36	23.89	3.24	2.90	-9.88	50.05	21.60
HB02	11.40	2.11	17.61	2.94	11.66	2.95	62.28	32.30
a) G3 modified	12.40	2.03	28.61	3.76	3.69	9.66	36.91	23.95
b) average	12.27	1.81	24.27	2.87	11.55	19.03	48.47	34.84
c) best set HEW04	11.79	1.71	21.21	2.11	15.90	23.01	57.89	46.16

Table 3: Absolute and relative (compared to Aurora 2 baseline) recognition results for feature prototype sets with changed envelope width, where filter functions exhibit only one maximum. a) Modified feature set G3sn b) Average over a total of 7 newly generated feature sets and c) best prototype set from these generated filters. As comparison, error rates for G3, HB02 and the Aurora baseline are shown. Gray shading denotes the best result per column.

by 19.6 % relative to the baseline for multi train and 12.6 % relative to the baseline for clean train.

For multi-condition training and full testing, best absolute results are achieved with HB02, but best relative results are obtained with HEW04. This is no contradiction, since relative WERs are computed by averaging over all conditions *after* calculating the reduction in error rate for each condition, so feature sets can be better in terms of absolute WER, but perform worse in terms of error rate reduction.

For the clean test condition, all previously evaluated feature prototype sets perform worse than the MFCC baseline. This is true for multi- and clean-trained systems, but does not affect average results, as the clean condition test is not incorporated in average results. HB02 for example yields 16.9 and 20.6 % relative increase in error for multi / clean condition training (see table 22 for the detailed results).
Sets with filters with only one maximum perform better in clean condition: For HEW04 relative reduction in WER is 13.84 / 4.96 % for multi- and clean-condition training, respectively (table 24).

Feature sets with only one maximum show superior performance in high SNRs, but this comes at the cost of reduced robustness. The strict constraints regarding the envelope width are accompanied by a simpler structure of modulation filters. The complexity is obviously not necessary and even detrimental in the absence of noise signals. However, in adverse acoustical conditions, the additional parameters ν_n and ν_k introduced in section 4.5 increase complexity and have a beneficial effect.
From a physiological point of view, it seems that the variability of receptive fields can not be modeled by the modified filters as well as with previously used filter functions. Inhibitory regions in the STRF are important when it comes to solving more complex problems like recognizing noisy speech, but for some of the filters in the set HEW04 inhibitory regions are hardly observable (as shown in Figure 11), and it might be that robustness is affected by this.

4.7 Fully Separable Filter Functions

The 2D-Fourier transform of a spectro-temporal receptive field (STRF) introduced in subsection 2.1.1 is called its transfer function. STRFs can be categorized by the properties of the transfer function:

- Quadrant-separable: The transfer function *within each quadrant* can be described as the outer product of a function of ω_k and a function of ω_t, i.e. the modulation frequencies in frequency and time direction.
- Fully separable: The complete transfer function can be described as the outer product of a function of ω_k and a function of ω_t. This implies, that the STRF can be fully described by the product of a spectral function with a temporal function. (Körding et al., 2001).
- Non-separable: The transfer function is neither quadrant- nor fully separable, i.e. it is an arbitrary, but complex conjugate symmetric, function in dependency of spectral modulation frequency ω_k and temporal modulation frequency ω_n.

It is estimated that 1/3 to 2/3 of the STRFs of neurons in the primary auditory cortex are fully separable and the remaining STRFs are quadrant separable. The LSTF filters were designed as quadrant separable functions as shown in Figure 12 a. Note that quadrant-separable functions generally have energy present in all four quadrants, but this is not the case for the spectro-temporal filters, that are fully-directional, so energy is only present in two opposing quadrants.

To account for this distribution found in physiology, separable modulation filters were designed as

$$sep(n,k) = h(n,k) \cdot f_1(2\pi\omega_k k) \cdot f_2(2\pi\omega_n n) \tag{10}$$

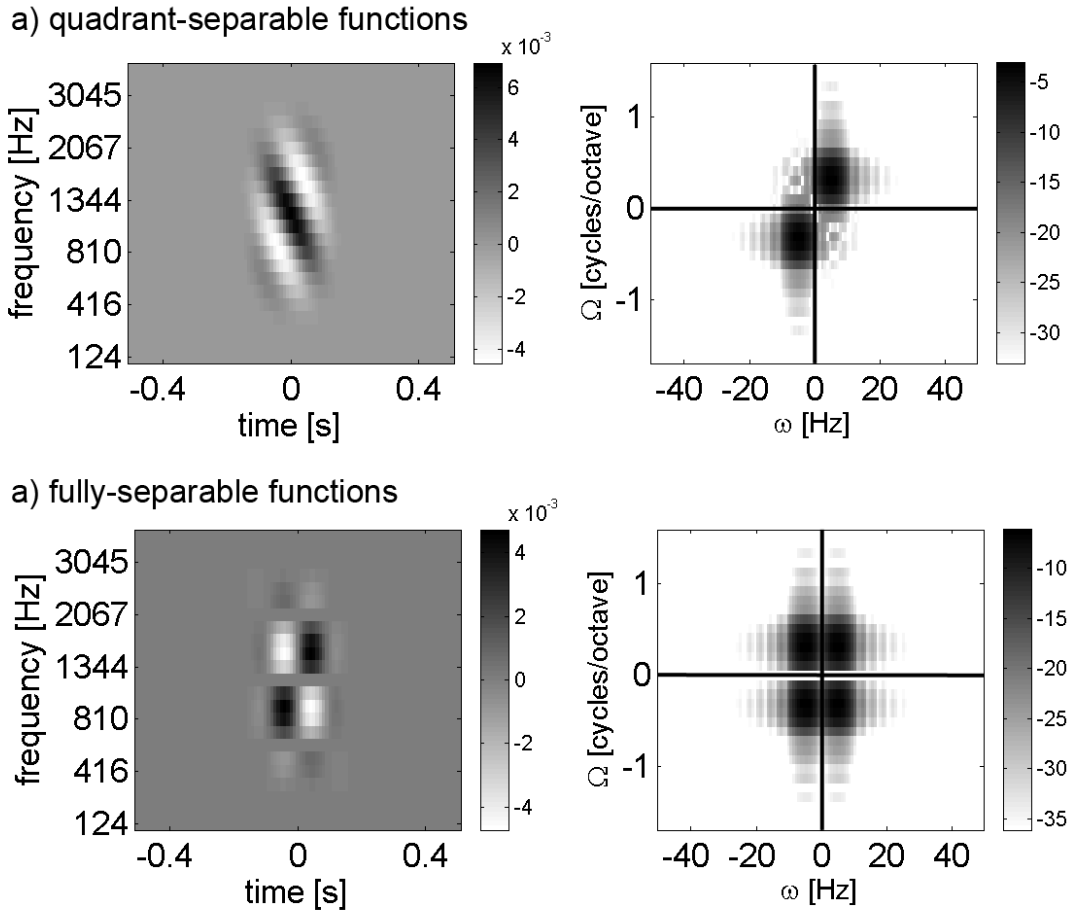

Figure 12: Quadrant-separable and fully-separable functions in time-frequency and modulation-frequency domain

where each of the functions f_1 and f_2 was substituted with either the sinus- or cosine function, which results in four 2-dimensional base functions. The Hanning envelope $h(n, k)$ was calculated according to equation 3. An example for such a function in time-frequency domain as well as in modulation-frequency domain is depicted in Figure 12 b. Limited spectro-temporal processing is possible with this type of features, as can be seen in this Figure, where upward moving ripples can be detected because the maxima form a diagonal structure.
As for previously investigated LSTF filters, the FFNN was used to determine a set of separable functions with parameters suitable to detect ASR-relevant information. The same physiological constraints as in section 2.1.2 were applied, so temporal and spectral modulation frequencies ω_n and ω_k ranged from 2 to 50 Hz and from 0.06 to 0.5 cycles/octave. The number of periods ν_x lay in the range 1..3.
Except for the new filter sets, the standard setup was not changed. Seven feature prototype sets were calculated with the FFNN, one of them being shown in Figure 13. All sets were evaluated with the HTK system.

Overall performance of the fully separable LSTF features is not as good as with best quadrant-separable filters. In average, error rates for multi-condition training are *increased* by 1.9 % relative (compared to Aurora 2 baseline). In contrast to this, clean-training results can be improved by 43.5 % relative, which means more robustness compared to the cepstral coefficients, but less performance in noisy conditions compared to

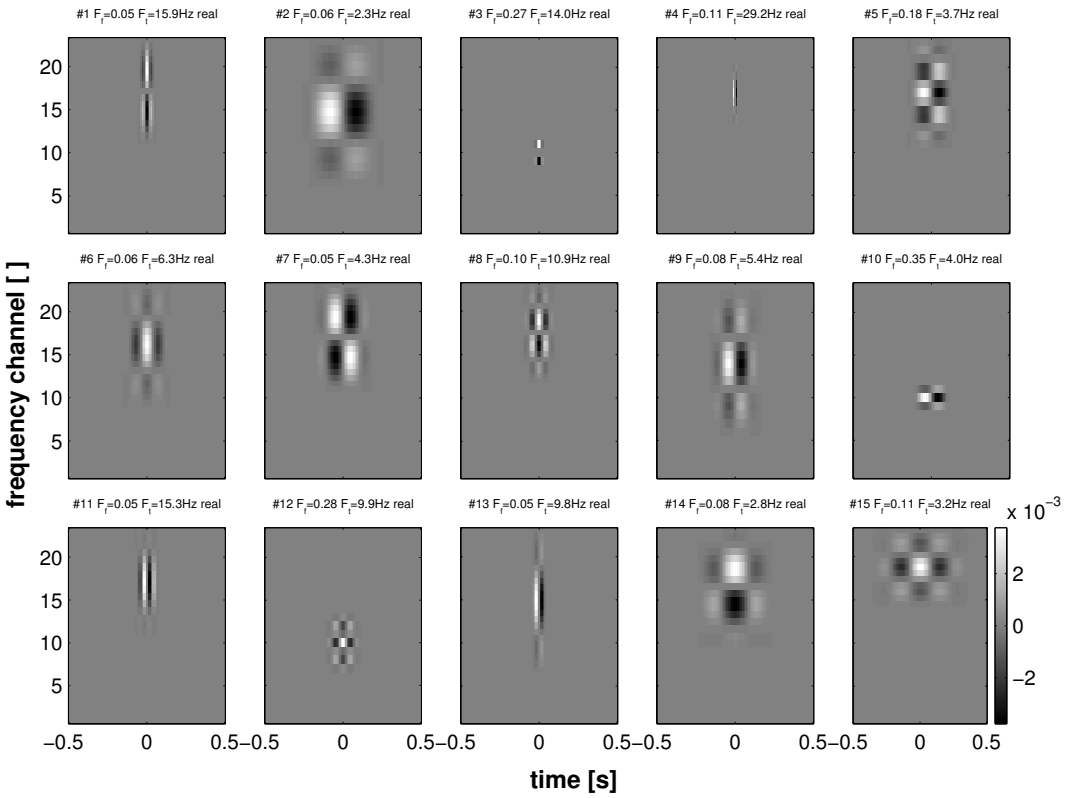

Figure 13: Separable filter functions in time-frequency domain. The 15 most important filters from the set with best performance are shown here.

other LSTF features.
The set with best performance (named SEP06) shows better WERs in average than G3. Relative improvements in WER compared to G3 results are depicted in Figure 14. These are presented in dependency of the Aurora noise signals. Subway M and Street M denote the noises used in test C, where a mobile phone frequency characteristic was applied to the speech and noise signals. Results are shown for the clean training condition only, for which the average relative improvement was 9.1 %. For multi-condition training relative improvements range from -15 to +14 % with an average increase of error rates of 4.3 %. The results for the noises "babble" and "restaurant" in Figure 14 are very noticeable: Absolute error rates compared to G3 are more than halved, from about 50 % to about 25 % WER for both noise conditions.
However, compared to the best sets with Hanning envelope, separable filter sets produce worse results in most conditions (with the noises "babble" and "restaurant" being an exception to this).

While overall performance of the new filter sets was not better than with the previously used LSTFs, separable filters have some properties worth discussing. Noise signals like "babble" and "restaurant" are the most difficult noises in ASR, as they exhibit the same long-term spectral properties as the speech to be recognized. Especially in these most adverse conditions, features derived from separable filters show improved performance compared to cepstral coefficients and to all other LSTF features tested so far.
It seems that the limited spectro-temporal processing the filters are capable of is not sufficient to deal with a large variety of noise types. However, the good performance in specific noise conditions suggests a combination with the previously used LSTF features

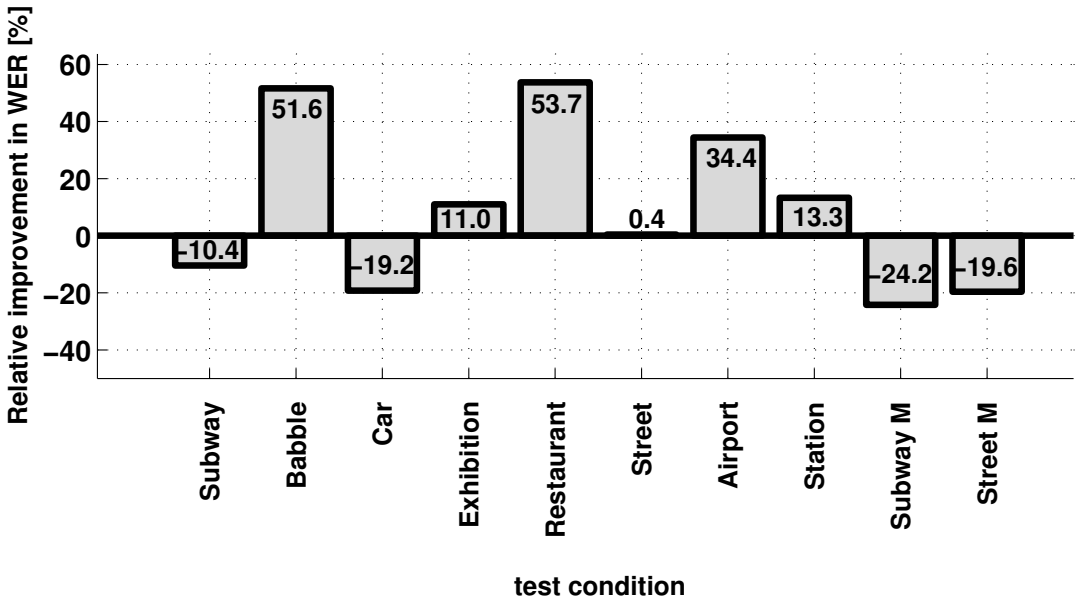

Figure 14: Relative improvement of WER compared to G3 for feature prototype set SEPB6. Results were obtained with an HTK system trained on clean-condition data. Overall performance of separable filters cannot compete with best LSTFs, but they show superior performance in some of the most adverse conditions.

to increase overall robustness. As neurons in A1 are likely to perform different filter operations, this also is reasonable from a physiological point of view.

4.8 Summary

In experiments presented in (Kleinschmidt, 2002a) and (Kleinschmidt, 2002c) 60 feature vector components derived from localized, spectro-temporal filters (LSTFs) were used and concatenated with deltas and double-deltas, yielding a vector dimensionality that is quite high compared to standard features like coefficients derived from mel-scaled cepstras or perceptual linear prediction.
The analyses regarding the number of features and necessity of deltas performed in this chapter reveal, that such high feature vector dimensionality is not necessary. With 20 features and deltas, i.e. a reduction from 180 to 40 vector components, a more robust feature extraction than with the Aurora 2 standard frontend can still be achieved. This is almost identical to the MFCC feature vector dimension in the Aurora 2 baseline setup, where 13 cepstral coefficients plus deltas and double deltas yield a 39 dimensional vector. A good compromise between recognition performance and computational cost is a feature vector with 50 components and single-deltas. This does reflect the available computing power rather than the physiological constraints: In the primary auditory cortex, thousands of neuronal detectors are present, so following the biological example would require thousands of feature vector components, which is not feasible with today's computer systems.

In section 4.5 it was shown, that Hanning-shaped localized, spectro-temporal filters (LSTFs) show sharper modulation frequency characteristics and therefore lead to increased performance compared to baseline results and feature sets with Gaussian envelope. This modification of the filter sets was thus used in all other following experiments. With other feature types no improvements compared to the best set with Hanning en-

velope have been achieved in general. The newly designed filters however show superior performance in different test conditions are valuable in specific applications: LSTF features, for which the number of maxima was limited to one, perform very well in high-SNR conditions and should be used when, e.g., close-talk microphone data is available. The opposite is true for features derived from fully separable LSTFs, that should be chosen for ASR system that have to deal with speech-like noise and are trained on clean-condition data only.
Separable filters can handle speech-like noise types very well, but deteriorate average performance. Thus, filters with diagonal structures are superior (in general) to separable functions. By using the latter, spectro-temporal information can be exploited, but not to the same extent as with non-separable LSTFs, which is evidence for the importance of spectro-temporal processing.
The experiments regarding envelope width indicate that limiting the number of maxima in LSTFs increases performance in clean and very high SNR conditions, while deteriorating performance for low SNR. A possibly reason for this is the lack of complexity (compared to other LSTF filters), as discussed in section 4.6.

A combination of these proposed filters with previously used filter prototypes promises increased overall performance: In order to decrease error rates, feature prototype sets could be composed of both filter types which can be achieved by allowing automatic selection of previously used LSTFs and fully separable functions in FFNN training. As mentioned earlier, in primary auditory cortex a mixture of neurons with fully- and quadrant-separable STRFs is present, so a combination of both filter types is physiologically reasonable.

5 Investigation of LSTF features with a State-of-the-Art System

The results in the previous section demonstrate the increase in robustness for features derived from localized, spectro-temporal filters (LSTFs) compared to mel-scaled cepstral coefficients (MFCCs). This section discusses the question, if these results that were obtained with a small-footprint system and a small vocabulary recognition task are scalable to a more complex state-of-the-art back end and to corpora containing mid- and large-sized vocabulary.

Neither for different corpora nor for classifiers of different complexity scalability is a trivial issue. An example for this are the results obtained by Hermansky et al. (2000), where improvements with the Tandem system have been achieved in a digit-recognition experiment, but not for a task where conversational speech had to be recognized.

A similar situation was observed for human speech recognition: Extensive speech recognition experiments with human listeners revealed that the importance of third octave frequency bands varies in dependency of the test corpus: For sentences, low frequency bands are more important for speech intelligibility than higher bands. This can be explained be the fact that in sentences missing phonemes can be completed using context. Since most missed phonemes in human speech recognition are high frequency consonants, context replaces the importance of high frequency bands. For single words the opposite was found, because no context can be used and the high frequencies have to be understood correctly. An example for this is shown in Figure 15 which shows the band importance functions of the Speech Intelligibility Index (SII) according to the ANSI standard (S3.5, 1997), where the importance of frequency bands is shown for two corpora, one containing conversational speech, the other one short words from a diagnostic rhyme test.

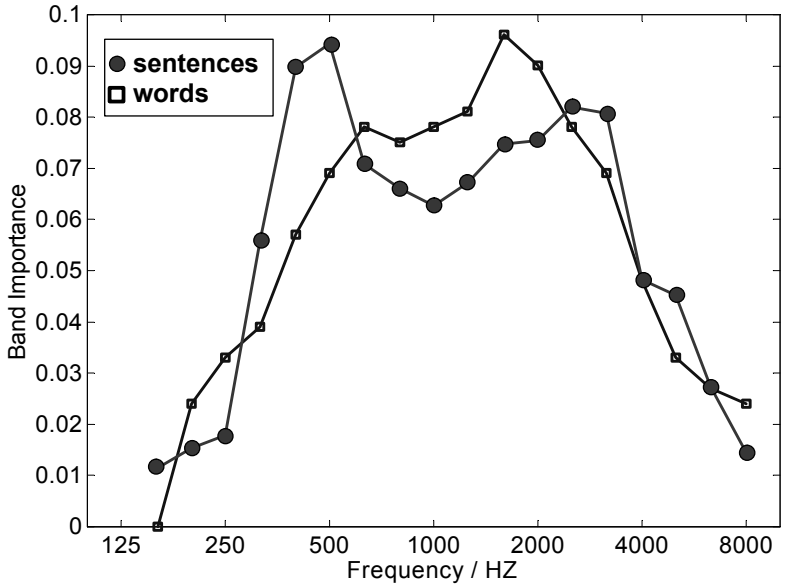

Figure 15: Importance of frequency bands for speech intelligibility. The importance depends on the content of the test material: For words without semantic context higher frequency bands are more important than lower bands (and vice versa for sentences with semantic context). The data labeled as "sentences" corresponds to short passages of easy reading material; the data labeled as "words" was derived from the Diagnostic Rhyme Test (DRT). Results were taken from (ANSI S3.5, 1997)

Two design parameters that influence the complexity of a classifier are the number of Gaussian or Laplacian distributions used to model the emission probabilities in a GMM and the structure of the classifier, where either a phoneme-based or a whole-word recognizer can be used. The influence of the first parameter was investigated in section 5.3 and section 5.5 by comparing recognition performance for two ASR setups, for which only the number of distributions differed. A phoneme-based recognizer was used in section 5.8 in conjunction with the CarCity corpus (instead of the whole-word models, that are employed for small vocabulary tasks as Aurora 2). The transferability to different speech databases was analyzed in the same section by testing LSTF features with corpora containing small to large-sized vocabulary and real-world recordings.

A second point of investigation was the complementarity of cepstral coefficients and LSTF features. This was done because of relatively poor results obtained with LSTF features in conjunction with a state-of-the-art HMM (section 5.3). A thought experiment in section 5.4 deals with the question, if MFCCs and LSTF features carry complementary information, so that a ASR setup with both feature types combined would be reasonable. This hypothetical experiment was motivation for a stream-combination setup [5] , which is presented in section 5.5.

Finally, it was analyzed by what means the overall performance of LSTF features in combination with a state-of-the-art system can be increased. The features were therefore tested as direct input to to a GMM-HMM backend and in combination with a MLP (section 5.3) in stream-combination with enhanced MFCCs (section 5.5). Additionally, beneficial effects of linear discriminant analysis (section 5.6) and noise suppression algorithms (section 5.7) were also investigated.

The ASR experiments presented here were carried out at the Philips Research Laboratories in Aachen. At Philips, a highly sophisticated state-of-the-art back end described in section 5.1 and advanced feature extraction methods (section 5.2) are used for research.

5.1 Description of the ASR system ASPIRIN

The Philips ASR system is called `ASPIRIN`, which is an acronym for "Advanced SPeech recognIzer for Research and INnovation". It is based on modules implemented in `C++` that are combined and controlled via a set of parameter files. Each module can be tested as stand-alone version. As the modules can handle data streams, data can be computed simultaneously, where communication between the modules is handled via the pvm (parallel virtual machine). This setup makes extraction, training and recognition very flexible and efficient at the same time.

The back end is highly optimized on training and recognition using MFCC features with several noise suppression methods applied (see 5.2) and uses Laplacian distributions instead of the more commonly used Gaussian distributions to create phoneme and word models. Discriminative training and maximum likelihood training are supported, whereas for our experiments the latter was employed.
The number of densities used to model the PDFs for a setup using 24-dimensional feature vectors was either 1867 or 14958, which we will refer to as tiny or full system. The benefit of the tiny system is faster training and recognition, but results in decreased complexity of the back end.

[5] If only one feature type is used as input to the back end, this is called single stream setup; if two or more feature types are combined (e.g. by concatenation) then this is referred to as multi-stream or stream-combination setup.

5.2 ASPIRIN Feature Extraction

The features commonly used with the Philips recognizer are mel-frequency cepstral coefficients (MFCCs). Feature extraction stage yields 12 cepstral coefficients plus 12 delta derivatives for each time frame. An important difference to the HTK setup used before is the frame shift of 16 ms introduced in the feature extraction stage. This doesn't deteriorate performance significantly (Lieb and Fischer, 2001), but made a conversion of LSTF prototype sets and the MLP training procedure necessary.
Aurora 2 baseline results show that without any further noise suppression, MFCCs are quite unrobust features. Therefore, a number of techniques are used to improve the feature extraction stage. A schematic overview of the extraction process is given in Figure 16. In the following, a short description of the applied enhancements is presented:

nonlinear spectral subtraction (NSS) removes additive noise from the signal. Let $S(t, f)$ denote the speech spectrum envelope corrupted by additive noise and $\hat{N}(t, f)$ be an estimate of the noise spectrum, obtained during noise-only periods. The subtraction rule is

$$\hat{X}(t, f) = max(S(t, f) - \alpha(t, f)\hat{N}(t, f), \beta\hat{N}(t, f))$$

with a time- and frequency-dependent overestimation factor $\alpha(t, f)$ that is determined from the current signal and noise condition. The floor factor β ensures a minimum noise floor in case the local noise estimate is larger than the current local speech plus noise signal. The noise estimate is obtained with a voice activity detector (VAD), that classifies a frame as speech or speech & noise (Lieb and Fischer, 2001).

noise masking (NM) as proposed in (Van Compernolle and Claes, 1996) is a technique used to remove some of the artifacts introduced by spectral subtraction and simulates the masking properties of the human auditory system. The goal is to normalize the SNR in each frequency band by adapting the masking constant depending on the measured SNR or dynamic range in each band. To achieve this, a masking function $M(t, k)$ is added to the filter bank energies $F(t, k)$ for each frame:

$$\overline{F}(t, k) = F(t, k) + M(t, k)$$

The instantaneous dynamic range of the masked signal SNR_I is determined and the masking constants are adapted in dependency of a fixed target dynamic range SNR. $M(t, k)$ is increased if $SNR_I(t) > SNR$ and decreased otherwise. Thus, the target SNR is tracked.

long term normalization (LTN) is used to remove slowly changing channel disturbances or convolutive noise. Each feature vector is filtered with a first-order high-pass filter. The long-term mean $\hat{\nu}_C$ of the cepstral features $C(t, k)$ is estimated by

$$\hat{\nu}_C(t, k) = \alpha\hat{\nu}_C(t - 1, k) + (1 - \alpha)C(t, k)$$

and then subtracted

$$\bar{C}(t, k) = C(t, k) - \hat{\nu}_C(t, k)$$

feature autoscaling : This technique is used to save storage capacity and computational cost. The value range of the features is linearly mapped to the range $[-127, 128]$, so features can be stored in the format `int8`. The data larger than the 99 percent quantile and smaller than the 1 percent quantile is clipped. The p

% quantile is the value L_p, for which p % of the observations is smaller and (100 - p) % is larger than L_p. The usage of quantiles is more reliable than the scaling according to minima and maxima, because statistical mavericks can be better compensated for . Because of this numerical dynamics are limited, but experiments in (Lieb and Fischer, 2002) showed, that this has little effect on performance.

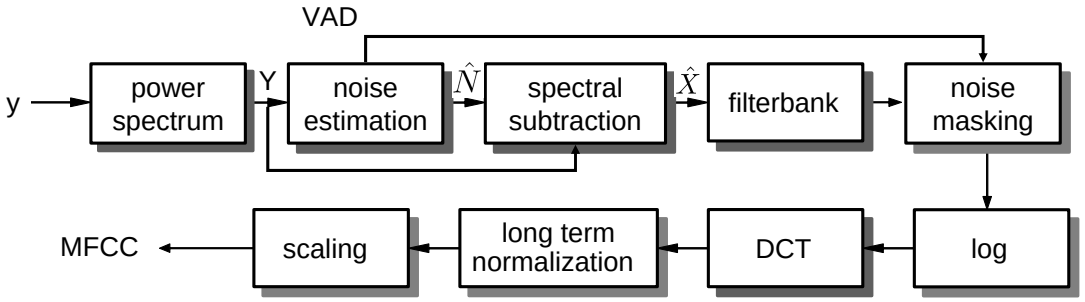

Figure 16: The noise robust MFCC front-end with spectral subtraction and noise masking, adapted from (Lieb and Fischer, 2002).

5.3 Aurora 2 - Single Stream

In this experiment, two questions are covered: Firstly, how do LSTF features perform in the `ASPIRIN` setup compared to the cepstral coefficients, that are usually employed? To answer this question, results were obtained with the enhanced MFCCs and chosen as new baseline; subsequently, recognition experiments with LSTF features as input to the `ASPIRIN` system were carried out. Secondly, how does a different number of parameters for the acoustical model (i.e. the complexity of the back end) affect performance? The experiments in this section and in section 5.5 were carried out with the tiny and the full setup to investigate this issue.

The Aurora 2 corpus was used as training- and test material (c.f. section 3) and the spectro-temporal features were either tested as direct input to the HMM or in conjunction with a MLP (analog to the setup with the HTK system depicted in Figure 6). Two LSTF prototype sets were selected to generate secondary features, namely the set that performed best with the HTK system (HB02) and the previously investigated set G3. The latter was chosen for reasons of comparability to previous experiments.
In the prototype feature sets, the values of temporal modulation frequency ω_n and standard deviation in temporal direction σ_n are given with respect to the frame shift. The prototype sets were optimized with mel-spectrograms with a 10 ms frame shift. To account for the 16 ms frame shift in the `ASPIRIN` setup, ω_n and σ_n were manually changed, so that the frequency characteristics are preserved with the new frame shift.
For experiments with the Tandem system, setup parameters were chosen as described in section 4.2, the only exception being the MLP, that was trained on TIMIT mel-spectras with 16 ms frame-shift and the corresponding adjusted phone-labels. The 24 most important PCA components (e.g. vectors corresponding to the largest eigenvalues) were used as input for the HMM, because the back end was tuned on 24-dimensional feature vectors, so providing more information by using all 56 PCA components gave worse error rates.
In all tests on Aurora 2, gender dependent models were used, where the information about the speaker's gender was derived from the corpus' filenames.
As comparison, the system was tested with and without the previously described noise

suppression methods. The results for MFCCs without NSS and NM were chosen as baseline. Relative improvements were calculated as $(WER_{Exp} - WER_{Base})/WER_{Base}$ (i.e. without calculating the relative improvement for each condition before averaging as described in section 4.1), since WER in dependency of SNR and noise condition was not available. Absolute and relative WERs in this single-stream setup are presented in table 4.

Tiny Setup

	absolute		relative	
	multi	clean	multi	clean
MFCCs (no NSS/NM)	20.25	49.84	0.00	0.00
LSTFs G3 (no MLP)	29.98	60.68	-48.05	-21.75
LSTFs HB02 (no MLP)	23.84	58.28	-17.73	-16.93
LSTFs G3 + MLP	14.72	27.45	27.29	44.92
LSTFs HB02 + MLP	12.90	19.18	36.30	61.52
MFCCs + NSS + NM	12.02	13.46	40.64	72.99

Full Setup

	absolute		relative	
	multi	clean	multi	clean
MFCCs (no NSS/NM)	16.28	45.11	0.00	0.00
LSTFs G3 (no MLP)	25.03	61.98	-53.75	-37.40
LSTFs HB02 (no MLP)	13.98	54.62	14.13	-21.08
LSTFs G3 + MLP	14.17	22.26	12.94	50.65
LSTFs HB02 + MLP	10.32	18.83	36.61	58.26
MFCCs + NSS + NM	8.60	10.19	47.17	77.41

Table 4: Absolute WER and relative WER improvement on Aurora 2 for MFCC and LSTF single stream setups. MFCC features without noise suppression have been chosen as baseline. Usage of a MLP greatly increases performance compared to the setup without MLP, but performance for LSTF features is worse than for MFCCs with noise suppression.

Results are consistent for the tiny and the full setup: Using LSTF features as direct input (i.e. without processing the data with a neural network) to the HMM produced very poor performance with about three times the error rates observed for enhanced MFCCs. The performance could be heavily increased by using the MLP prior to HMM processing. For the tiny setup, the usage of an MLP lowered absolute WER by 11 % for multi condition training and 37 % (!) for clean condition training. Addition of MLP-processing for the large-footprint recognizer yielded slightly smaller improvements with about 4 and 36 % for the multi- and clean-trained system, respectively. Even with the neural network, the best LSTF features showed much worse performance than MFCCs with NSS and NM applied (with 4.2 % higher WERs in average). This motivates a further investigation, by what means overall performance of LSTF features in conjunction with the state-of-the-art back end can be increased.

A reason for the poor performance without usage of the MLP might be a high degree of correlation and a disadvantageous distribution of LSTF features across the feature space. This hypothesis is supported by the fact that LDA (instead of MLP and an additional PCA) helps to improve results by about 8 percent absolute (statistics derived from clean-condition speech for clean training and from multi-condition data for multi training, respectively). Of course, benefits with the MLP are much larger, which indicates that the non-linear remapping of feature space is obviously much better suited to the distribution of LSTF features. It seems that the non-linear magnification of interesting

regions in feature space provoked by the MLP is especially important for LSTF features, possibly because these regions lie close together, so that variability contained in speech signals produces no large differences in position in the (non-transformed) feature space and hence complicate the recognition task for the back end.

The large difference between the best LSTF set and the enhanced MFCCs (especially for clean condition training) demonstrates the efficiency of the noise reduction methods that were applied to MFCCs. Therefore, the same techniques were applied to LSTF features (see section 5.7).

The fact that increasing the number of PCA components (and with it information of features) deteriorates performance shows that tuning is another problem we have to deal with. This is a difficulty that very often occurs with new feature types that are integrated in existing ASR systems: Changing the parameters of the system usually increases error rates because a tuned system resides in a local optimum, that is left when changes are made (Bourlard et al., 1996).

5.4 Do LSTFs and MFCCs carry complementary information?

The answer to this question could indicate if the combination of these feature types is reasonable: If features are complementary and therefore carry different recognition-relevant information, a combination is surely more indicated than a combination of features that are not complementary.

In order to answer the question, it was investigated to what extent an ASR system using LSTF features and a system using MFCCs produce the same errors. If for both systems similar sentences or words are correctly subscripted and similar errors occur, then the complementary information is small or inexistent. If on the other hand completely different words are erroneous, the complementary information can be regarded as large.

In the first case, the intersection I_E of the two sets L and M containing wrongly subscripted sentences or words would be almost identical to the smaller set (as shown in Figure 17 a). For the second case, the intersection of the sets would be small or empty (Figure 17 b). The number of elements contained in I_E in relation to the total number of elements is equal to the SER or WER associated with I_E.

To obtain I_E we carried out a thought experiment (gedankenexperiment), where an imaginary oracle determines *before* a sentence or word was processed, which ASR system (the one with LSTF or the one with MFCC features) will produce less errors. With this *a priori* knowledge, we chose the system with better performance for the current sentence or word. Errors that occur despite the oracle-knowledge are errors that are produced by both systems. Thus, the set of these errors is identical to I_E. This was achieved by analyzing the recognition results for both an ASR system using MFCCs and a system with LSTF features. With this results at hand, performance increase by using the oracle was analyzed. The reduction in error rate by using the oracle is a measure of complementarity.

The thought experiment can be varied with respect to the knowledge the oracle has: One can either employ the decision of the oracle on a sentence or on a word level. In order to identify the oracle selection on sentence level, it was determined for each sentence if MFCCs or LSTFs produce an error. If both or neither lead to an error, the MFCC system is arbitrarily selected. If the MFCC sentence is erroneous, but the LSTF sentence correctly subscripted, the LSTF system is selected and vice versa. An example is shown in Figure 18. The number of erroneous sentences selected with oracle-knowledge divided

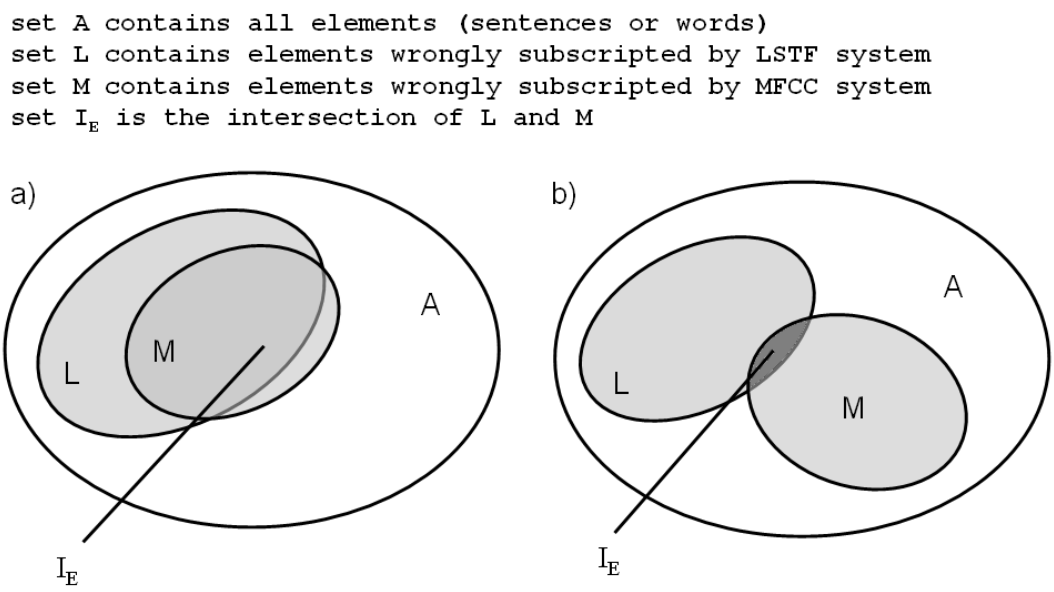

Figure 17: The number of elements in the intersection I_E of M and L is a measure for complementarity of two feature types. Systems with few (a) or much (b) complementary information are symbolized.

by the total number of sentences is the oracle sentence error rate SER_{I_E}. A sentence was regarded as erroneous if either a word was inserted or a existing word was not or wrongly subscripted.

The oracle selection on word level was determined the following way (see Figure 18): If both streams produce an error (insertion, deletion or substitution) at the same position in the sentence, then this is counted as an error in the "predicted" sentence. The oracle word error rate WER_{I_E} is then determined according to equation 9.

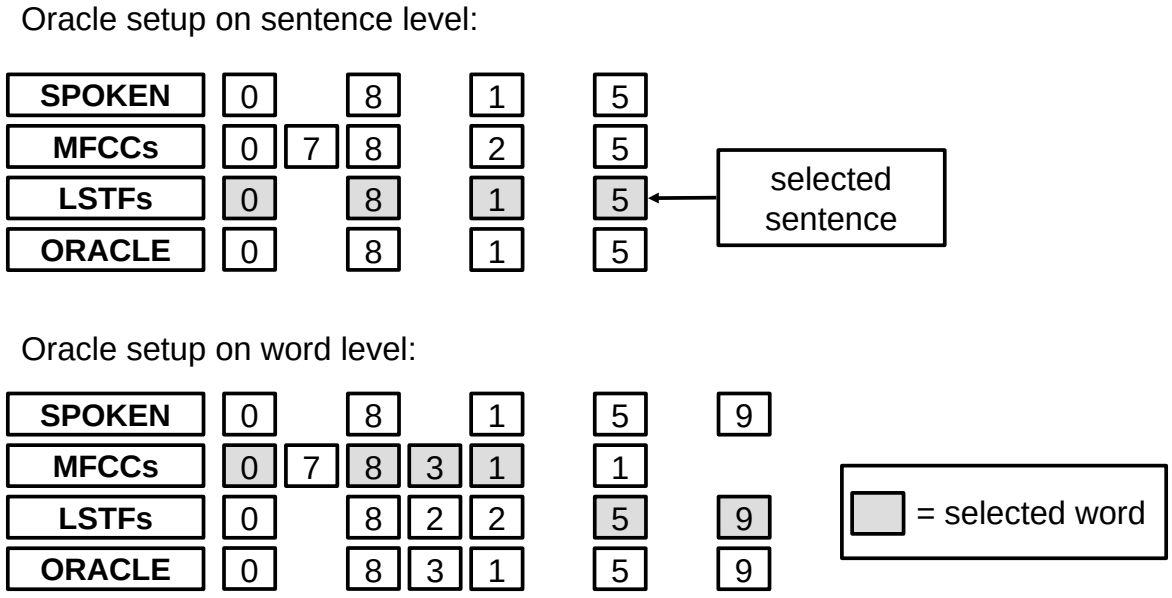

Figure 18: Examples for oracle experiment a) on sentence level, where a sentence is selected, if it is correctly recognized and the other one produces an error b) on word level, where this is done for every word. In the lower part of the figure, gray shading denotes words that are selected for the oracle system.

Furthermore, for the two feature types it was investigated how word errors were distributed among the eleven target classes. Different distributions are another evidence for complementarity.

MFCC features				
	Set A	Set B	Set C	Overall
Multi	18.08	18.95	21.47	19.50
Clean	21.41	21.30	23.06	21.92
Average	19.75	20.13	22.27	20.71

MFCC features				
	Set A	Set B	Set C	Overall
Multi	7.95	10.07	10.79	9.60
Clean	10.01	8.55	10.01	9.52
Average	10.01	9.31	10.40	9.56

LSTF features				
	Set A	Set B	Set C	Overall
Multi	19.55	23.05	24.14	22.25
Clean	32.00	40.04	29.78	33.94
Average	25.78	31.55	26.96	28.09

LSTF features				
	Set A	Set B	Set C	Overall
Multi	9.21	22.14	15.20	15.52
Clean	17.33	10.73	11.72	13.26
Average	17.33	16.44	13.46	14.39

Oracle				
	Set A	Set B	Set C	Overall
Multi	5.70	5.13	6.74	5.86
Clean	5.15	2.89	6.54	4.86
Average	5.43	4.01	6.64	5.36

Oracle				
	Set A	Set B	Set C	Overall
Multi	4.36	4.90	5.06	4.77
Clean	3.48	3.67	4.43	3.86
Average	3.92	4.29	4.75	4.32

Table 5: SER (left) and WER (right) for two different ASR systems using MFCC features (with NSS and NM) and LSTF features. A comparison with the results of the combined oracle system indicates features of highly complementary content.

5.4.1 Results

Results of the thought experiment are presented in table 5. Small oracle error rates alone do not prove complementarity. Only by comparing these numbers with recognition performance of the single stream systems complementarity can be estimated. Thus, error rates of the MFCC and LSTF system are presented beside the oracle performance. For the MFCC system noise masking and non-linear spectral subtraction were applied to the speech files. LSTF features were calculated using the prototype set HB02 and processed with a MLP before feeding them to the HMM. In each case, feature dimensionality was 24.
The overall SER can be decreased from 20.71 % (LSTF system) and 28.09 % (MFCC system) to 5.36 % when employing the oracle knowledge. Similar results are obtained for the analysis on word level, where an average relative improvement of 54.1 % is achieved for the oracle system (compared to the MFCC system).

In Figure 19 (p. 52) the distribution of error frequency over target class are shown for MFCC features (top) and LSTF features (bottom). The scale on the left denotes absolute word error frequency; relative error frequency is depicted on each bar. Results for LSTFs show a large deviation (relative results range from 5.4 to 17.3 % with a standard deviation of 3.44 %), while the distribution for MFCCs is more homogenous with a value range from 7.8 to 11.4 % and a standard deviation of 1.28 %.
For both distributions, the target class with the most errors is identical (namely the target class 'oh'). Consisting of only one phoneme, this is the shortest word in the corpus, which indicates that systems with both feature types have difficulties to model such short words, as only little contextual information is available.
On the other hand, quite dramatic differences between the results can be seen in other cases: For example, the target class 'zero' is above average for MFCCs, for the LSTF system, it is the second worst class. The target class 'eight' is recognized better by the LSTF system. This is true for relative and absolute values (report results here). Except

for the class 'oh', the best and the worst three classes for LSTF and MFCC features are disjunct.
Errors of recognition systems are randomly distributed to some extent. The MLP training for example is initialized with randomly chosen weights, so the training result is not deterministic. If errors were *completely* randomly distributed, the resulting error rate with oracle-knowledge would be equal (or almost equal) to the product of the two single-stream error rates. However, the inhomogeneous distribution in Figure 19 shows that errors occur systematically. The same is true for errors in dependency from the SNR, as for increasing SNR word error rates consistently drop. This is documented by the detailed Aurora 2 results in section 7. Nevertheless, SER for the thought experiment is still smaller than the product of the single sentence error rates.
All this shows that errors of the systems are very differently distributed. The usage of delta derivatives for MFCC features helps to incorporate some temporal dynamics contained in speech, but the spectro-temporal information inherent to the LSTF features is also an important factor. Thus, feature combination of MFCCs and LSTFs surely is beneficial.

5.5 Aurora2 - Stream Combination

In the previous section it was shown, that a combination of cepstral coefficients and spectro-temporal features is promising. Therefore, the experimental setup was modified, so that feature streams were combined by concatenation, yielding 48-dimensional vectors that were processed by the back end. The feature streams were generated according to section 5.2 and 5.3. Combined features correspond to the extraction stage depicted in Figure 20 (3). For the full system setup, the word penalty variable (WP) was varied, i.e. a parameter that is used for HMM training and basically controls the ratio between misses and insertions.

Error rates were obtained with prototype sets G3 and HB02. Results for the G3-MFCC stream were consistent with results from the G3 single stream setup, i.e., performance was slightly worse than with the improved HB02 filter set, so no detailed results are reported here. Absolute and relative results for both the tiny and the full setup are shown in table 6. For the tiny setup, error rates can be substantially improved by feature concatenation. Absolute error can be decreased by 2.9 % in average, compared to the enhanced MFCC results. This corresponds to an averaged relative improvement of 18.7 %. Best results with the full system were achieved with $WP = 0$ and $WP = 5000$, which either gave better performance for the clean-trained or the multi-trained HMM, so results for both variants are shown. The full system evaluation yields very similar results as the tiny HMM, with an average relative improvement of 15.3 %.

The results demonstrate the scalability of error rates, when the number of model parameters is changed: Error rates can be consistently lowered with the full setup. This is true for for all feature types and training conditions. Furthermore, the relative improvements between the two setups are very similar. From this we can conclude, that improvements achieved with the tiny setup are likely to be transferable to the full setup. This is an advantage because training and decoding with a less complex model is much faster. As an example, training and recognition with the tiny setup took about 2 hours, and between one and two days with the full system. Moreover, the results demonstrate that the gap between the LSTF-MFCC-Concatenation system on the one hand and the oracle system on the other hand is quite large (8.00 % for the real system, 4.32 % for the oracle system). Improved stream combination techniques might help to increase performance

Tiny Setup

	absolute		relative	
	multi	clean	multi	clean
MFCCs (no NSS/NM)	20.25	49.84	0.00	0.00
MFCCs + NSS + NM	12.02	13.46	40.64	72.99
LSTFs HB02 + MLP	12.90	19.18	36.30	61.52
Combined Features	9.82	10.89	51.51	78.15

Full Setup

	absolute		relative	
	multi	clean	multi	clean
MFCCs (no NSS/NM)	16.28	45.11	0.00	0.00
MFCCs + NSS + NM	8.60	10.19	47.17	77.41
LSTFs HB02 + MLP	10.32	18.83	36.61	58.26
Combined Features, wp = 5000	6.62	9.34	59.34	79.30
Combined Features, wp = 0	6.88	9.11	57.74	79.80

Table 6: Multi-stream results on the Aurora 2 paradigm with the ASPIRIN setup. Results are shown for the tiny and the full system setup, wp=0 and wp=5000 denote different values for the word penalty variable. Relative improvement compared to MFCCs without noise suppression are presented. Feature streams were generated as depicted in Figure 20 (3).

by enhanced exploitation of complementary information, e.g. with an additional MLP, a PCA or a LDA. The latter was investigated in the following section.

5.6 Decorrelation and Reduction of Dimensionality

When doubling the number of features per time frame, the size of the covariance matrix of the HMM has to be adapted. For a diagonal covariance matrix that was used here, the number of parameters increases linearly with input vector dimensionality. Higher number of parameters is an advantage when modeling probability density functions, so performance is likely to be increased, which is documented by the results for the tiny and the full system in section 5.3.
In order to find out if the performance gain observed in the multi-stream setup is due to increased information or simply due to a more complex back end, feature vector dimensionality was reduced with LDA. As MFCCs and LSTFs are calculated independently, the degree of correlation between the feature streams is also reduced by using LDA, potentially yielding an increase in accuracy.

Statistics for LDA were estimated from clean training data for the clean-trained back end, and analogously for the multi-trained back end (matched statistics-training condition). Class information was obtained with a forced alignment, which is also used in systems for automatic phoneme-labeling. The different classes correspond to single states of the whole-word models. No frame-context was used for the LDA training. The 48-dimensional, concatenated features were projected in a 24-dimensional subspace.
For a fair comparison, MFCC features were also decorrelated by LDA, where the number of 24 vector components was not reduced. As for LSTF features, statistics were derived from the Aurora 2 training corpus. Temporal contextual information was included by concatenating each feature vector $C(n)$ with the preceding vector $C(n-1)$ and the following vector $C(n+1)$.

For reasons of time, effects of LDA were only investigated with the tiny system setup. Error rates are presented in Figure 5.6. One of the main goals of using LDA and noise suppression (see next section) was to investigate whether benefits can be achieved compared to the best feature extraction commonly used with `ASPIRIN`. Hence, the improved cepstral coefficients with NSS and NM are used as baseline.

	absolute WER		relative improvement	
	multi	clean	multi	clean
a) MFCCs + NSS + NM	12.02	13.46	0.00	0.00
b) MFCCs + NSS + NM + LDA	11.87	13.37	1.25	0.67
c) LSTFs HB02 + MLP	12.90	19.18	-7.32	-42.50
d) a + c	9.82	10.89	18.30	19.09
e) a + c + LDA	9.19	10.69	23.54	20.58

Table 7: Comparison of absolute and relative performance for Aurora 2 setups with and without LDA. MFCCs with NSS and NM applied were chosen as baseline. For the stream combination setup (d), LDA reduces feature vector dimensionality and increases performance at the same time (e). Feature streams were calculated according to Figure 20 (4).

LDA enhances performance for MFCCs only very little, with relative improvements of 1.3 % and 0.7 %. For the system with combined feature streams, benefits are larger: Compared to the stream setup without LDA, the decorrelation results in additional 5 % and 2 % relative improvement for the multi- and clean-trained system (or 0.6 % and 0.2 % difference in terms of absolute WER).
Improvements for LDA are only available for the tiny system setup, so a direct comparison with the oracle WERs is not reasonable. A look on the relative improvements shows, that LDA is the preferable feature combination technique compared to simple concatenation. Still, not all information inherent to the single-streams was utilized, since the average relative improvement with LDA (22.1 %) is far from the results from the thought experiment, where relative improvement was more that 50 %. Consequently, further optimization of feature combination is likely to increase overall performance (which remains to be tested in future experiments).
Results show that the benefits in performance observed for the stream combination setup are due to a gain in information and cannot be explained with increased complexity of the HMM, because errors can be lowered in spite of the reduction of the number of vector components.
LSTF features and MFCCs are calculated from the same speech database and are subsequently concatenated. This might result in a higher degree of covariance between components of the combined vector compared to cepstral coefficients alone, which are calculated from different frequency components and therefore presumably exhibit a lower degree of covariance. This might be the reason for higher relative improvements obtained with the combined features and LDA (compared to MFCC features with LDA).

5.7 Noise Suppression Methods for LSTFs

For a fair comparison of MFCCs and LSTF features, noise spectral subtraction and noise masking, as well as long time normalization were applied to the Aurora 2 speech signals before computing LSTF features. The procedure was repeated for the Aurora-noised TIMIT corpus, that was subsequently used to train the MLP. Afterwards, the modified features were tested in a single-stream setup, in combination with MFCCs and with LDA (as depicted in Figure 20 (2-4)). Again, for reasons of time, results were only obtained

for the tiny setup.
As table 8 shows, the usage of noise suppression algorithms generally improved performance of LSTF features. In the single-stream setup, the largest performance gain can be observed with 1.2 % absolute decrease in WER. For the concatenated features, the improvement is statistically not significant with only 0.06 % reduction in terms of absolute WER, where the 95 % confidence interval was ±0.2 %. Finally, a LDA on top of the concatenated feature stream with NSS and NM brings about 5 % relative improvement (or 0.6 % absolute).
This setup gives the best results on Aurora 2, where WERs are reduced by 3.4 % absolute compared to the MFCC + NSS + NM baseline, which corresponds to a relative improvement of 26.9 %. For multi- and clean-condition training similar improvements are observed.

	Absolute Word Error Rates			Relative Improvement		
	Multi	Clean	Average	Multi	Clean	Average
a) MFCCs + NSS + NM	12.02	13.46	12.74	0.00	0.00	0.00
b) LSTFs	12.90	19.18	16.04	-7.32	-42.50	-25.90
c) Concatenation of a) and b)	9.82	10.89	10.36	18.30	19.09	18.68
d) like c) + LDA	9.19	10.69	9.94	23.54	20.58	21.98
e) LSTFs + NSS + NM	13.39	16.27	14.83	-11.40	-20.88	-16.41
f) Concatenation of a) and e)	10.03	10.56	10.3	16.56	21.55	19.15
g) like f) + LDA	9.03	9.6	9.31	24.88	28.68	26.92

Table 8: Comparison of LSTF performance with and without noise suppression techniques (single stream and combination). Single-stream performance can be increased by using NSS and NM (e). For stream combination, no significant improvement is observed (f). An additional LDA (g) brings about 5 % relative reduction compared to the same case without NSS and NM (d).

LSTF features were designed with robustness in mind, and in fact, setups with LSTFs do not profit from noise suppression methods to the same extent as cepstral coefficients do. As was shown in section 5.3, performance loss for MFCCs without noise suppression is 22.3 % *absolute*, compared to 1.2 % for LSTF features.
It is interesting that beneficial effects are observed for MLP output data and the combined features with subsequent LDA, but not for the stream-combination setup without LDA. A reason for this might be, that complementary information of the feature types was reduced because the same techniques to deal with adverse conditions were used.

5.8 Tests on CarDigits and CarCity

A current key application for ASR systems is operating a car navigation system (e.g. for a destination entry by voice) or car computer system (e.g. for voice and name dialing in conjunction with a mobile phone, or information retrieval). Due to the adverse noisy background conditions and the fact that only recordings from a far-field microphone are available, this is a challenging task, so robustness is a very desirable property of such systems. The large vocabulary needed for navigation system complicates the recognition task.

LSTF features have proven to perform considerably better than standard MFCCs as used in the Aurora 2 baseline setup. Furthermore, the results of the previous subsections demonstrate, that performance of a heavily tuned state-of-the-art recognizer in conjunction with enhanced cepstral coefficients can be increased in stream-combination setups. In the following it is investigated, whether the results achieved with the `ASPIRIN` Aurora 2 setup are transferable to different corpora with real-world recordings and to large-vocabulary tasks.

Thus, experiments with CarDigit and CarCity as described in section 3 were carried out. Depending on the test set that is used, CarCity can be categorized as a medium or large vocabulary corpus, so this is a new paradigm in the list of LSTF and Gabor features experiments, that have been tested on speech databases with fewer target classes (like Aurora 2) or in a very large vocabulary test (like conversational speech, as proposed in (Kleinschmidt and Gelbart, 2002)).
For both experiments, LSTF feature extraction was almost identical: From train and test data mel-spectrograms with 16 ms frame shift were calculated. The CarDigit sample rate was 16 kHz, but due to technical issues, filter bank outputs were limited to 4 kHz as with the HTK setup.
LSTF features were calculated with the sets HB02 and G3 adjusted to the 16 ms frame shift for the CarCity setup, for the CarDigit experiment only HB02 was used. Secondary features were processed with the MLP and decorrelated via PCA (see section 5.3). Additionally, for the CarCity setup performance for G3 *without* an MLP and subsequent PCA was tested. All LSTF feature vectors were limited to 24 components as for the Aurora 2 setup.
MFCC feature extraction for corpora with car-recordings differs from the techniques presented in section 5.2: Coefficients are computed from 16 kHz data, with filter bank center frequencies ranging from 316 to 7000 Hz. 12 MFCCs are used, which are combined with 12 first-order dynamic features in the case of the CarDigit setup. For CarCity, 11 delta-derivatives (energy plus 10 lowest quefrencies) and one double-delta feature (derived from energy component) are added. NSS and LTN were employed, but noise masking was not applied.
The back end in the digit recognition system was configured as for the Aurora 2 `ASPIRIN` setup that uses whole-word models, which means that a HMM model is trained for every class in the vocabulary. For medium- and large-sized vocabularies, phoneme-based HMMs are used, where each phoneme is modeled by a state sequence of a HMM, which has to be decoded in the recognition process. Because of the different structure, a phoneme-based back end can be regarded as more complex than a whole-word recognizer.

If the feature vectors are not well represented by the mixtures, the number of mixtures per state can be doubled in the training process, which is called density splitting. This procedure can be repeated several times, where each iteration yields a HMM with more parameters and hence more modeling power. The splitting criterion for the `ASPIRIN` system was based on thresholding of observation counts (Lieb and Fischer, 2001), i.e. densities are split if the dissimilarity between model and observation exceeds a certain threshold. In the CarDigit setup, the densities were split five times, which is referred to as 5-split system and corresponds to the Aurora 2 full system setup. For the city names setup, recognition performance for a 4-split, a 5-split and a 6-split system were obtained.
Cepstral coefficients, LSTF features and a combination of both (48 dimensions) were fed to the HMM back end. Word error rates for the CarDigit single- and multi-stream experiments are reported in table 9. Results for the CarCity corpus are shown in table 10.

	absolute WER [%]
a) MFCC features	6.66
d) LSTF HB02 + MLP	12.39
c) Concatenation of a) and b)	6.66

Table 9: Results for tests on CarDigit corpus. Word error rates for 12 MFCC features + 12 delta derivatives, 24 LSTF-MLP feature vector components, generated with feature set HB02, and the combination of these are presented.

Lexical size	splits		Single Stream (24 dim LSTF)		
		MFCCs + NSS	G3	G3 + MLP	HB02 + MLP
3k	4	26.17 ± 1.6	97.48 ± 0.6	95.50 ± 0.8	68.14 ± 1.7
	5	25.25 ± 1.6	97.17 ± 0.6	95.33 ± 0.8	63.95 ± 1.8
	6	24.36 ± 1.6	97.51 ± 0.6	94.55 ± 0.8	63.92 ± 1.8
10k	4	35.88 ± 1.8	99.25 ± 0.3	97.65 ± 0.6	83.03 ± 1.4
	5	35.20 ± 1.8	98.91 ± 0.4	97.85 ± 0.5	79.42 ± 1.5
	6	34.72 ± 1.8	99.18 ± 0.3	97.38 ± 0.6	78.50 ± 1.5
Lexical size	**splits**		**Multi Stream (MFCCs + LSTF features)**		
			G3	G3 + MLP	HB02 + MLP
3k	4		49.78 ± 1.8	30.39 ± 1.7	26.03 ± 1.6
	5		46.85 ± 1.8	29.71 ± 1.7	24.36 ± 1.6
	6		44.91 ± 1.8	28.82 ± 1.7	23.45 ± 1.6
10k	4		59.18 ± 1.8	41.77 ± 1.8	36.87 ± 1.8
	5		56.59 ± 1.8	41.19 ± 1.8	35.16 ± 1.8
	6		54.28 ± 1.8	40.82 ± 1.6	34.70 ± 1.6

Table 10: Absolute word error rates on CarCity corpus. Results are shown in dependence of corpus size (3k or 10k) and back end complexity (number of splits). LSTF features with and without MLP processing were tested in single- and multi-stream setups, combined with MFCCs.

In both experiments, performance was not improved with LSTF features, neither for the single- nor for the multi-stream setup.
For the CarDigit setup, error rates were approximately doubled when substituting the MFCCs with LSTF features. A concatenation of both produced exactly the same error rate as the MFCCs alone (although the error rate in dependency of the speaker was differently distributed).
For CarCity, improved MFCCs already yield very high error rates of about 25 % for the 3k corpus and 35 % for the 10k corpus. WERs for the LSTF single stream setup are much higher than this, the worst being G3 without neural network processing. In this case, absolute WERs are close to 97 % (3k lexicon) and 99 % (10k lexicon). As in previous tests, performance of HB02 is higher than G3, where a Gaussian envelope was used in the filter process. Differences between the two sets are very noticeable in this setup, which indicates that improved modulation frequency characteristics are more important in this particular acoustic situation. It is difficult to judge if this because the real-world recordings contain more convolutive noise than the Aurora 2 corpus of due to the stationarity of the car-noises. Hence, further work with additional speech material should be conducted.
Results are markedly improved for the stream combination setup, but for G3 (with or without MLP) in combination with MFCCs no improvement compared to the MFCC sys-

tem alone can be achieved. As for the CarDigit setup, a combination of MLP-processed HB02 features and MFCCs yields almost identical results as MFCCs alone.
As expected, performance increases with the number of splits. Largest benefits are observed for a 5-split system, which shows that this is a good compromise between performance and computational cost.

Apart from the tuning of the back end, which was addressed in the previous section, a number of reasons for the poor performance can be specified:
Firstly, the average SNR for both corpora was about 10 dB. A closer look on the detailed results for set G3 (see table 25) and HB02 (table 22) reveals, that LSTF features perform not as good as MFCCs without noise suppression at this particular SNR and in this type of noise. Relative performance compared to the HTK Aurora baseline for G3 is -15.45 % and -7.85 % for HB02. Relative differences to MFCCs *with* NSS and NM is estimated to be even higher.
For cepstral coefficients, the spectral information up to 8 kHz was used in the experiments. Due to technical issues the filter bank outputs for LSTF feature extraction had to be limited to 4 kHz; the rest of the information was discarded. In the case of car noise, disturbances in the low frequency bands outweigh noises in higher frequency bands, so this is a disadvantageous condition for the LSTF system. From experience, the performance gain by doubling the sample rate of the CarCity speech data can be quantified with about 10-15 % for MFCCs, when German corpora where used as in our tests. For English corpora, improvements from higher sample rates are not as large.
Finally, the MLP was trained with TIMIT mixed with Aurora 2 noises, that contain stationary car-noise, but also several other noise types. This large variation of noise types might have a detrimental effect on performance in these particular setups.

In order to improve recognition performance, a number of measures are suggestive: First, for the feature selection process data with a sampling rate of 16 kHz instead of the present 8 kHz could be used, by choosing a different training corpus than zifkom. Combined with an application of car-noise instead of Aurora noises, filter functions could be determined that exploit more spectral information and lie in frequency bands, where disturbances by car noise are not as serious.
Second, digit recognition experiments showed that LSTF features benefit from noise suppression techniques and improved feature combination methods like LDA, so applying these would probably lower average error rates.
Additionally, training material for the MLP could be improved by substituting the Aurora noises, that were mixed with TIMIT, with car-only noises. Optimally, the Car training corpora should be used, but this would require phoneme-labeling of the whole corpus. Here a drawback of the tandem setup becomes apparent, that was already pointed out in (Hermansky et al., 2000): The task specifity is very high for an MLP, even when it is trained with such different noise types as provided with the Aurora 2 corpus, so changing the recognition task makes a new configuration of the neural network necessary.

5.9 Summary

In this section, features derived from filter sets as suggested in (Kleinschmidt, 2002b) as well as improved filter sets introduced in section 4 were evaluated with the Philips state-of-the-art recognition system `ASPIRIN`.
One aim was to show that results that previously obtained results are transferable to

different recognition tasks and to more complex recognition systems. This aim was only partly achieved:

For the corpora CarDigit and CarCity, no beneficial effects of LSTF feature in combination with MFCCs with noise spectral subtraction were observed. In section 5.8 the reasons are discussed and a number of possible solutions are presented.
In the CarCity setup, a phoneme-recognizer with a more complex structure than the previously used whole-word-model HMMs was used. Since no improvements have been achieved in this scenario, scalability of performance to this kind of recognizer was not verified. The similar results in the CarCity and the CarDigit setup however suggest, that the rather poor WERs are caused by the corpus properties and not by the structure of the back end.

The results demonstrate the scalability of error rates, when the number of model parameters is changed: Error rates can be consistently lowered with the full setup. This is true for for all feature types and training conditions. Furthermore, the relative improvements between the two setups are very similar. From this we can conclude, that improvements achieved with the tiny setup are likely to be transferable to the full setup. This is an advantage because training and decoding with a less complex model is much faster. As an example, training and recognition with the tiny setup took about 2 hours, and between one and two days with the full system.

Error rates obtained with back ends of different complexity show that LSTF features profit from higher number of model parameters. The consistency of relative improvements for the tiny and the full system setup indicates, that results scale with model complexity. Thus, beneficial effects observed for the tiny system are presumably present in the full system setup, which is an important result because of the large gap in training and decoding time between back ends of different complexity.

A further goal was the demonstration of complementarity of MFCCs and LSTF features: In the Aurora 2 paradigm, LSTF features in a single-stream setup show worse performance than cepstral coefficients with NSS and NM. The combination of features was motivated in section 5.4, where complementarity of the MFCCs and LSTFs was documented. In spite of the back end being tuned on cepstral coefficients, relative improvements of 17 % average relative improvement compared to MFCCs *with* NSS and NM were achieved.

The overall performance of the Philips recognizer was further enhanced by application of advanced feature transformation and noise suppression algorithms, which yielded a relative improvement of 27 %, where the combined feature stream had the same dimensionality as the reference MFCC feature vectors. Tuning the recognition system to this new type of feature stream is likely to produce even higher improvements.

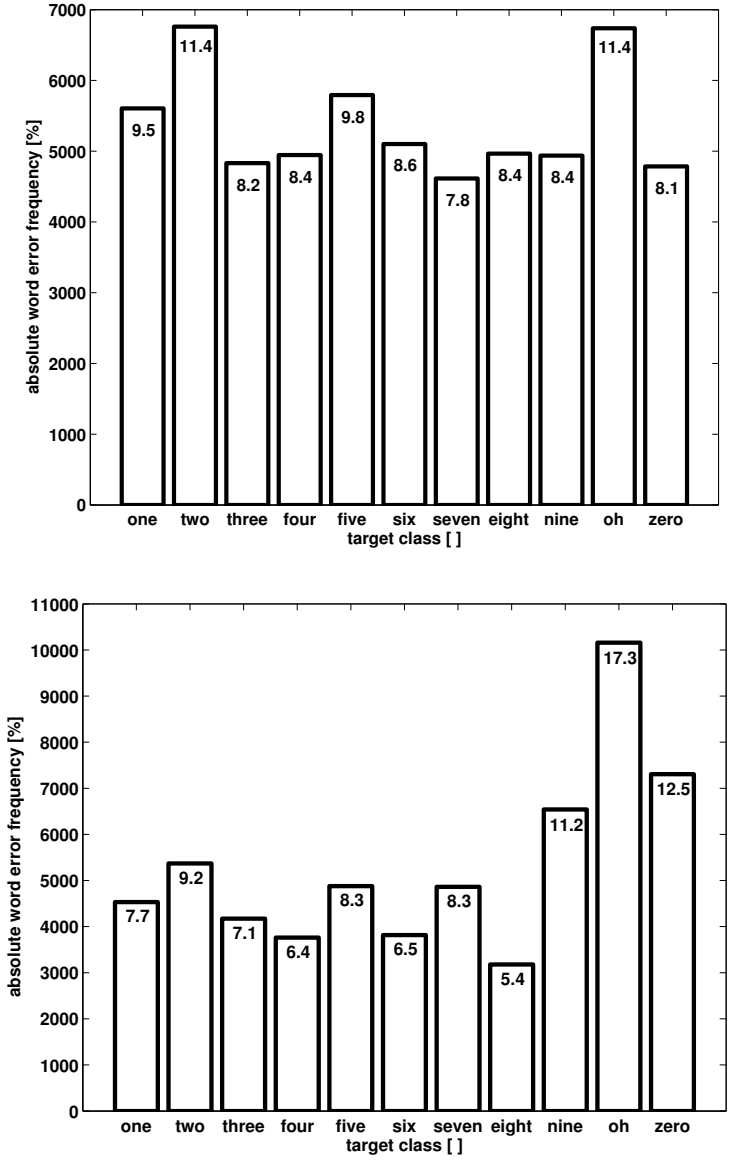

Figure 19: Distribution of absolute word errors over target classes for MFCC (top) and LSTF (bottom) features. The labels on each bar denote the relative word error per class.

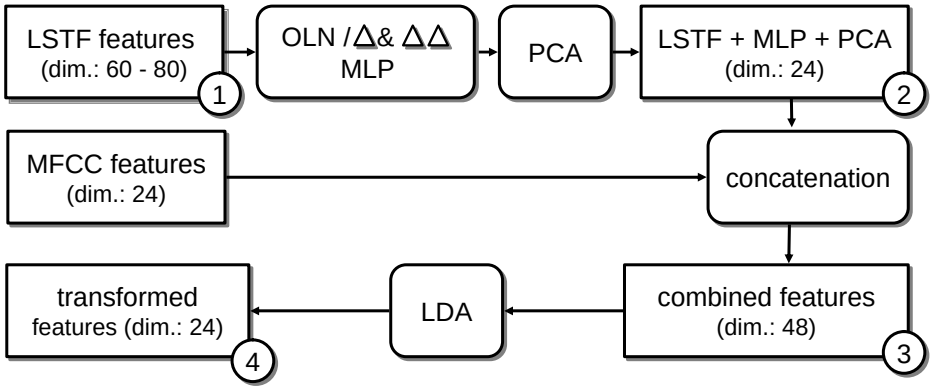

Figure 20: LSTF features (1) are processed with an MLP. The MLP output is decorrelated and its dimension is reduced to 24 with a PCA (2). MFCC features are generated according to Figure 16. By concatenation these are combined with LSTF-MLP output. The result is a 48 dimensional feature vector(3), whose dimensionality may be reduced with a subsequent linear discrimant analysis (LDA),(4).

6 Overall Summary & Conclusion

In this thesis, a number of experiments are presented that analyze the previously proposed Gabor-shaped LSTFs and investigate methods of improvement regarding overall performance and robustness of ASR systems.

In section 4 it was shown that acceptable performance can be achieved, when the number of elements is the same as for the widely used cepstral coefficients, eliminating a major point of criticism for LSTF features. Still, robustness of such low-dimensional feature vectors is better compared to the Aurora 2 baseline.

The changes to Gabor features motivated by knowledge from signal-processing yields improved feature sets with higher performance than the reference set G3. The use of Hanning envelopes gave constantly improved results, that were confirmed with both the HTK setup as well as with the state-of-the-art system, for which relative improvements of over 40 % were observed relative to the G3 reference set.

Further physiological constraints produced prototype sets which also performed better than G3, but overall performance was not as high as with the sets where only the envelope was substituted. However, these filter sets exhibit superior performance in specific conditions, either in high SNRs or in most adverse conditions like babble and restaurant noise. A combination of different filter types promises a further increase of recognition accuracy.

While superior robustness of LSTF features with mid-sized corpora containing car-recordings could not be approved, a number of improvements are suggested, that are likely to deliver better performance than the baseline results obtained with MFCC features with noise suppression. This should be investigated in future experiments.
Word error rates obtained with different `ASPIRIN` setups indicate that results with LSTF features scale with the number of parameters used to model the emission distributions of the HMM states.

The analysis regarding complementarity shows that beneficial effects can be expected by combining cepstral coefficients and LSTF features, and in fact, up to 27 % relative improvement was achieved with a stream-combination setup and an untuned state-of-the-art system.

These results clearly demonstrate that speech processing benefits from auditory modeling in speech processing and that it is worthwhile to integrate information over time *and* frequency on the feature level. It was shown that a parametric filter function such as the localized, spectro-temporal features are capable of this task.

A further goal of research should be an investigation of the dependency of the ASR system: Performance of Gabor- and LSTF-features with a GMM-HMM system is very poor compared to the Tandem system. A statistical analysis of LSTF features could help to find out why non-linear processing as with the MLP is so important in a LSTF setup.

The feature combination experiments with the `ASPIRIN` recognizer motivate future work regarding stream combination: The results of the oracle-thought experiment show that error rates can be drastically lowered, when the information inherent to both feature streams is optimally exploited. The combination of multi-stream features and LDA yielded superior performance compared to enhanced MFCCs alone, but the oracle-error rates are still much better, so that a thorough optimization of stream combination appears reasonable.

Two of the largest benefits of LSTF features are the feature prototype set selection with a neural network and the complementarity to MFCCs (and probably to other standard feature extraction techniques that rely on spectral information only) due to spectral-temporal processing. These advantages could be combined when the relevance of feature prototypes were determined in context with other features, so that only filters are selected that yield maximal complementary information.

7 Annex

7.1 Detailed Results

In this section detailed Aurora 2 results, obtained with the HTK system are presented. This includes absolute error rates and improvements relative to the Aurora 2 baseline system.

Multicondition training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	1.47	1.48	1.58	1.76	1.57	1.47	1.48	1.58	1.76	1.57	1.35	1.57	1.46	**1.55**
20 dB	2.00	1.93	1.76	2.16	1.96	2.21	2.03	1.91	1.76	1.98	1.72	2.09	1.91	**1.96**
15 dB	2.64	2.54	2.59	2.84	2.65	3.50	2.66	2.86	2.59	2.90	2.86	3.20	3.03	**2.83**
10 dB	4.91	4.90	4.12	5.49	4.86	6.23	4.44	4.83	5.21	5.18	5.13	5.20	5.17	**5.05**
5 dB	10.65	10.76	9.75	11.76	10.73	14.58	12.76	11.01	12.71	12.77	12.04	13.06	12.55	**11.91**
0 dB	30.12	34.25	37.10	31.13	33.15	37.46	34.43	31.40	38.97	35.57	37.73	40.15	38.94	**35.27**
-5dB	72.37	76.63	80.64	70.29	74.98	74.55	73.19	72.08	79.39	74.80	77.06	76.42	76.74	**75.26**
Average	10.06	10.88	11.06	10.68	**10.67**	12.80	11.26	10.40	12.25	**11.68**	11.90	12.74	**12.32**	**11.40**

Clean training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	1.07	1.09	1.28	0.99	1.11	1.07	1.09	1.28	0.99	1.11	1.04	1.15	1.10	**1.11**
20 dB	1.81	1.87	2.09	2.25	2.01	2.46	2.36	2.36	1.82	2.25	2.49	2.51	2.50	**2.20**
15 dB	3.35	3.02	2.68	3.98	3.26	6.39	3.90	3.70	3.42	4.35	3.50	4.35	3.93	**3.83**
10 dB	7.18	9.49	5.91	8.02	7.65	14.06	7.74	9.54	7.37	9.68	7.18	8.40	7.79	**8.49**
5 dB	16.86	27.45	16.22	18.64	19.79	33.47	21.16	25.29	21.66	25.40	16.46	21.19	18.83	**21.84**
0 dB	40.16	62.76	49.36	47.70	50.00	66.53	47.88	56.99	53.13	56.13	42.71	49.91	46.31	**51.71**
-5dB	74.46	92.05	85.54	81.15	83.30	94.47	78.26	87.44	84.54	86.18	77.34	78.51	77.93	**83.38**
Average	13.87	20.92	15.25	16.12	**16.54**	24.58	16.61	19.58	17.48	**19.56**	14.47	17.27	**15.87**	**17.61**

Figure 21: Absolute Aurora 2 word error rates for prototype feature set HB02

Multicondition training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	-4.26%	-11.28%	-10.49%	-50.43%	-19.11%	-4.26%	-11.28%	-10.49%	-50.43%	-19.11%	2.17%	-18.05%	-7.94%	**-16.88%**
20 dB	8.26%	6.31%	0.00%	14.62%	7.30%	2.64%	15.06%	20.08%	24.79%	15.64%	26.18%	13.64%	19.91%	**13.16%**
15 dB	21.19%	1.17%	-12.61%	8.97%	4.68%	8.85%	20.12%	15.88%	30.56%	18.85%	18.29%	13.28%	15.78%	**12.57%**
10 dB	12.63%	-8.17%	-7.85%	6.79%	0.85%	11.76%	13.62%	23.21%	14.31%	15.72%	16.86%	14.47%	15.66%	**9.76%**
5 dB	3.09%	8.74%	21.81%	5.16%	9.70%	2.47%	4.92%	11.71%	14.35%	8.36%	28.72%	17.55%	23.13%	**11.85%**
0 dB	6.31%	6.98%	19.17%	14.22%	11.67%	4.24%	6.79%	6.91%	7.06%	6.25%	29.86%	8.02%	18.94%	**10.96%**
-5dB	1.46%	-5.45%	-1.08%	7.96%	0.72%	-2.28%	-1.18%	-2.84%	-1.46%	-1.94%	4.61%	-1.53%	1.54%	**-0.18%**
Average	10.30%	3.01%	4.10%	9.95%	6.84%	5.99%	12.10%	15.56%	18.21%	12.97%	23.98%	13.39%	18.69%	**11.66%**

Clean training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	3.60%	-12.37%	-34.74%	-33.78%	-19.32%	3.60%	-12.37%	-34.74%	-33.78%	-19.32%	-25.30%	-26.37%	-25.84%	**-20.63%**
20 dB	44.31%	80.23%	28.42%	40.79%	48.44%	75.05%	43.00%	76.52%	65.07%	64.91%	62.44%	48.46%	55.45%	**56.43%**
15 dB	60.45%	89.14%	76.59%	60.08%	71.57%	74.92%	67.09%	85.86%	81.77%	77.41%	74.95%	60.13%	67.54%	**73.10%**
10 dB	70.66%	81.89%	83.79%	71.06%	76.85%	70.78%	77.20%	81.19%	83.55%	78.18%	74.41%	66.36%	70.39%	**76.09%**
5 dB	67.98%	64.39%	76.58%	69.40%	69.59%	54.28%	66.79%	66.46%	71.14%	64.67%	66.66%	57.13%	61.90%	**66.08%**
0 dB	48.22%	33.57%	44.72%	44.37%	42.72%	28.37%	42.07%	36.32%	41.29%	37.01%	43.41%	34.64%	39.02%	**39.70%**
-5dB	16.66%	7.84%	8.19%	12.88%	11.39%	4.62%	14.36%	7.69%	9.93%	9.15%	11.21%	11.60%	11.40%	**10.50%**
Average	58.32%	69.84%	62.02%	57.14%	61.83%	60.68%	59.23%	69.27%	68.56%	64.44%	64.37%	53.34%	58.86%	**62.28%**

Figure 22: Relative Aurora 2 word error rates for prototype feature set HB02.

Multicondition training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	0.98	1.12	1.49	1.05	1.16	0.98	1.12	1.49	1.05	1.16	1.07	1.18	1.13	**1.15**
20 dB	1.32	1.36	1.88	1.82	1.60	1.41	1.69	1.52	1.57	1.55	1.38	1.93	1.66	**1.59**
15 dB	1.75	1.96	2.42	2.56	2.17	2.18	2.96	2.33	2.41	2.47	2.49	2.75	2.62	**2.38**
10 dB	4.11	4.20	4.21	5.52	4.51	5.50	5.32	4.29	5.12	5.06	5.59	5.62	5.61	**4.95**
5 dB	10.32	10.67	10.59	12.56	11.04	14.61	13.88	11.93	12.77	13.30	13.51	15.27	14.39	**12.61**
0 dB	29.14	36.46	40.11	30.58	34.07	40.65	37.33	33.07	40.14	37.80	43.02	43.89	43.46	**37.44**
-5dB	69.51	77.03	80.35	67.82	73.68	78.57	74.52	73.40	80.50	76.75	79.89	79.08	79.49	**76.07**
Average	9.33	10.93	11.84	10.61	**10.68**	12.87	12.24	10.63	12.40	**12.03**	13.20	13.89	**13.55**	**11.79**

Clean training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	0.92	0.76	1.01	0.86	0.89	0.92	0.76	1.01	0.86	0.89	0.77	0.82	0.80	**0.87**
20 dB	2.09	1.69	1.82	2.47	2.02	1.57	2.18	1.61	1.42	1.70	2.09	2.57	2.33	**1.95**
15 dB	3.41	3.05	2.80	4.50	3.44	4.08	3.63	2.98	2.90	3.40	3.53	4.32	3.93	**3.52**
10 dB	7.65	10.25	7.49	9.13	8.63	12.40	9.31	9.16	8.73	9.90	8.75	9.43	9.09	**9.23**
5 dB	19.50	35.16	27.41	22.25	26.08	37.67	27.12	30.75	30.58	31.53	22.69	28.78	25.74	**28.19**
0 dB	46.58	74.33	73.04	52.79	61.69	72.83	60.82	65.88	66.21	66.44	56.74	62.55	59.65	**63.18**
-5dB	77.89	93.95	90.22	82.66	86.18	93.34	83.83	87.83	88.58	88.40	83.70	84.19	83.95	**86.62**
Average	15.85	24.90	22.51	18.23	**20.37**	25.71	20.61	22.08	21.97	**22.59**	18.76	21.53	**20.15**	**21.21**

Figure 23: Absolute Aurora 2 word error rates for prototype feature set HEW04.

Multicondition training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	30.50%	15.79%	-4.20%	10.26%	13.09%	30.50%	15.79%	-4.20%	10.26%	13.09%	22.46%	11.28%	16.87%	**13.84%**
20 dB	39.45%	33.98%	-6.82%	28.06%	23.67%	37.89%	29.29%	36.40%	32.91%	34.12%	40.77%	20.25%	30.51%	**29.22%**
15 dB	47.76%	23.74%	-5.22%	17.95%	21.06%	43.23%	11.11%	31.47%	35.39%	30.30%	28.86%	25.47%	27.17%	**25.98%**
10 dB	26.87%	7.28%	-10.21%	6.28%	7.56%	22.10%	-3.50%	31.80%	15.79%	16.55%	9.40%	7.57%	8.48%	**11.34%**
5 dB	6.10%	9.50%	15.08%	-1.29%	7.35%	2.27%	-3.43%	4.33%	13.95%	4.28%	20.01%	3.60%	11.81%	**7.01%**
0 dB	9.36%	0.98%	12.61%	15.73%	9.67%	-3.91%	-1.06%	1.96%	4.27%	0.31%	20.02%	-0.55%	9.74%	**5.94%**
-5dB	5.35%	-6.00%	-0.71%	11.20%	2.46%	-7.79%	-3.01%	-4.72%	-2.88%	-4.60%	1.10%	-5.06%	-1.98%	**-1.25%**
Average	25.91%	15.10%	1.09%	13.35%	13.86%	20.31%	6.48%	21.19%	20.46%	17.11%	23.81%	11.27%	17.54%	**15.90%**

Clean training, multicondition testing														
	A					B					C			
	Subway	Babble	Car	Exhibition	Average	Restaurant	Street	Airport	Station	Average	Subway M	Street M	Average	**Average**
Clean	17.12%	21.65%	-6.32%	-16.22%	4.06%	17.12%	21.65%	-6.32%	-16.22%	4.06%	7.23%	9.89%	8.56%	**4.96%**
20 dB	35.69%	82.14%	37.67%	35.00%	47.62%	84.08%	47.34%	83.98%	72.74%	72.04%	68.48%	47.23%	57.85%	**59.43%**
15 dB	59.74%	89.03%	75.55%	54.86%	69.80%	83.99%	69.37%	88.61%	84.54%	81.63%	74.73%	60.40%	67.57%	**74.08%**
10 dB	68.74%	80.44%	79.46%	67.05%	73.92%	74.23%	72.58%	81.94%	80.51%	77.32%	68.82%	62.23%	65.53%	**73.60%**
5 dB	62.97%	54.39%	60.42%	63.48%	60.31%	48.54%	57.44%	59.22%	59.25%	56.11%	54.04%	41.78%	47.91%	**56.15%**
0 dB	39.94%	21.32%	18.20%	38.44%	29.47%	21.59%	26.41%	26.39%	26.84%	25.31%	24.82%	18.09%	21.45%	**26.20%**
-5dB	12.83%	5.94%	3.17%	11.26%	8.30%	5.76%	8.26%	7.27%	5.63%	6.73%	3.90%	5.20%	4.55%	**6.92%**
Average	53.42%	65.46%	54.26%	51.77%	56.23%	62.48%	54.63%	68.03%	64.78%	62.48%	58.18%	45.95%	52.06%	**57.89%**

Figure 24: Relative Aurora 2 word error rates for prototype feature set HEW04. Note the improved error rates for clean condition test, compared to the feature set with best overall performance, HB02, shown on page 58

Aurora 2 Small Vocabulary	Multicondition training, multicondition testing													
		A					B					C			
Detailed relative results in terms of error reduction. Halving the error rate = +50%		Subway	Babble	Car	Exhibition	Average	Restauran	Street	Airport	Station	Average	Subway M	Street M	Average	Average
	Clean	4.26%	-38.35%	-12.59%	-40.17%	-21.71%	4.26%	-38.35%	-12.59%	-40.17%	-21.71%	2.17%	-45.11%	-21.47%	**-21.66%**
	20 dB	35.32%	-2.91%	-1.70%	12.25%	10.74%	-1.32%	7.53%	22.59%	34.19%	15.75%	44.64%	4.96%	24.80%	**15.55%**
	15 dB	34.03%	-26.85%	-1.30%	2.24%	2.03%	10.42%	3.00%	13.24%	23.86%	12.63%	26.29%	13.28%	19.78%	**9.82%**
	10 dB	11.57%	-35.54%	-15.45%	0.51%	-9.73%	-8.36%	-14.01%	22.26%	6.58%	1.62%	15.40%	15.95%	15.68%	**-0.11%**
	5 dB	-2.27%	-14.08%	21.09%	-1.05%	0.92%	-17.66%	2.46%	9.62%	11.46%	1.47%	30.20%	21.21%	25.70%	**6.10%**
	0 dB	8.62%	-4.83%	24.05%	12.51%	10.09%	-6.26%	9.58%	2.55%	11.11%	4.25%	40.70%	19.18%	29.94%	**11.72%**
	-5dB	7.52%	-6.95%	2.69%	9.99%	3.31%	-5.09%	2.50%	-2.64%	1.51%	-0.93%	13.00%	5.18%	9.09%	**2.77%**
	Average	17.45%	-16.84%	5.34%	5.29%	2.81%	-4.64%	1.71%	14.05%	17.44%	7.14%	31.44%	14.92%	23.18%	**8.62%**

Aurora 2 Small Vocabulary	Clean training, multicondition testing													
		A					B					C			
Detailed relative results in terms of error reduction. Halving the error rate = +50%		Subway	Babble	Car	Exhibition	Average	Restauran	Street	Airport	Station	Average	Subway M	Street M	Average	Average
	Clean	-8.11%	-27.84%	-47.37%	-87.84%	-42.79%	-8.11%	-27.84%	-47.37%	-87.84%	-42.79%	-25.30%	-59.34%	-42.32%	**-42.69%**
	20 dB	31.08%	48.84%	28.42%	22.89%	32.81%	39.25%	19.57%	70.35%	47.22%	44.09%	61.54%	29.16%	45.35%	**39.83%**
	15 dB	49.59%	52.93%	67.16%	34.10%	50.95%	37.83%	53.59%	75.38%	68.07%	58.72%	65.71%	48.76%	57.24%	**55.31%**
	10 dB	55.33%	42.01%	74.64%	45.87%	54.46%	33.11%	56.38%	64.03%	67.97%	55.37%	60.05%	52.78%	56.42%	**55.22%**
	5 dB	52.49%	25.33%	57.57%	42.76%	44.54%	20.34%	49.44%	44.07%	51.64%	41.37%	55.28%	45.38%	50.33%	**44.43%**
	0 dB	31.32%	10.37%	26.45%	21.05%	22.30%	8.93%	25.35%	18.79%	22.61%	18.92%	42.35%	31.47%	36.91%	**23.87%**
	-5dB	10.69%	-2.48%	5.83%	-3.84%	2.55%	-5.33%	7.54%	1.10%	5.00%	2.08%	15.01%	12.15%	13.58%	**4.56%**
	Average	43.96%	35.90%	50.85%	33.34%	41.01%	27.89%	40.86%	54.52%	51.50%	43.69%	56.99%	41.51%	49.25%	**43.73%**

Figure 25: Detailed, relative Aurora 2 word error rates for prototype feature set G3.

7.2 List of abbreviations

- A1 - Primary Auditory Cortex
- ANN - Artificial Neural Network
- ASPIRIN - Advanced SPeech recognIzer for Research and INnovation
- ASR - Automatic Speech Recognition
- FFNN - Feature Finding Neural Network
- HMM - Hidden Markov model
- HATS - Hidden Activation TRAPS
- HTK - HMM Toolkit
- ICSI - International Computer Science Institute
- KLT - Karhunen-Loéve Transformation
- LDA - Linear Discriminant Analysis
- LSTF - Local Spectro Temporal Filters
- LTN - Long Term Normalization
- ML - Maximum Likelihood
- MLP - Multi Layer Perceptron
- NSS - Non-linear Spectral substraction
- OLN - Online Normalization of mean and variance
- PCA - Principal Component Analysis
- PLP - Perceptual Linear Prediction
- RASTA - RelAtive Spectral TrAnsformation
- SER - Sentence Error Rate
- SNR - Signal-to-Noise Ratio
- TRAPS - TempoRAl PatternS
- VAD - Voice Activity Detector
- WER - Word Error Rate

References

Adami, A., Burget, L., Dupont, S., Garudadri, H., Grezl, F., Hermansky, H., P. Jain, S. K., Morgan, N. and Sivadas, S. (2002). QUALCOMM-ICSI-OGI features for ASR. In *Proc. ICSLP*.

ANSI S3.5 (1997). Methods for calculation of the speech intelligibility index.

Bourlard, H., Hermansky, H. and Morgan, N. (1996). Towards increasing speech recognition error rate. *Speech Communication*, **18**:205–231.

Bourlard, H. and Morgan, N. (1998). Hybrid HMM/ANN systems for speech recognition: Overview and new research directions. In *Adaptive Processing of Sequences and Data Structures*, Vol. 1387 of *Lect. Notes in AI*, pp. 389–417. Giles, C.L. and Gori, M.

Chen, B., Chang, S. and Sivadas, S. (2003). Learning discriminative temporal patterns in speech: Development of novel traps-like classifiers. In *Eurospeech*.

De-Valois, R. and De-Valois, K. (1990). *Spatial Vison*. Oxford U.P., New York.

deCharms, R. C., Blake, D. T. and Merzenich, M. M. (1998). Optimizing sound features for cortical neurons. *Science*, **280**:1439–1443.

Depireux, D., Simon, J., Klein, D. and Shamma, S. (2001). Spectro-temporal response field characterization with dynamic ripples in ferret primary auditory cortex. *J. Neurophysiol.*, **85**:1220–1234.

Elhilali, M., Klein, D., Fritz, J. and Shamma, S. (2003). What does precise spiking in ai tell us about the structure of its receptive fields? In *ARO*.

Ellis, D. and Gelbart, D. (2004). Icsi speech faq. *http://www.icsi.berkeley.edu/speech/faq/*.

Gramss, T. (1991). Fast algorithms to find invariant features for a word recognizing neural net. In *IEEE 2nd International Conference on Artificial Neural Networks*, pp. 180–184. Bournemouth.

Gramss, T. and Strube, H. W. (1990). Recognition of isolated words based on psychoacoustics and neurobiology. *Speech Communication*, **9**:35–40.

Hermansky, H. (1998). Should recognizers have ears? *Speech Communication*, **25**:3–24.

Hermansky, H., Ellis, D. and Sharma, S. (2000). Tandem connectionist feature extraction for conventional HMM systems. In *ICASSP*.

Hirsch, H. and Pearce, D. (2000). The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions. In *ISCA ITRW ASR*.

Jain, P. and Hermansky, H. (2003). Beyond a single critical band in TRAP-based ASR. In *Eurospeech*. submitted.

Kaernbach, C. (2000). Early auditory feature coding. In *Contributions to psychological acoustics: Results of the 8th Oldenburg Symposium on Psychological Acoustics.*, pp. 295–307. BIS, Universität Oldenburg.

Kleinschmidt, M. (2002a). Methods for capturing spectro-temporal modulations in automatic speech recognition. *Acustica united with acta acustica*, **88**:416–422.

Kleinschmidt, M. (2002b). Robust speech recognition based on spectro-temporal features. *Carl von Ossietzy University Oldenburg*.

Kleinschmidt, M. (2002c). Spectro-temporal Gabor features as a front end for ASR. In *Proc. Forum Acusticum, Sevilla*.

Kleinschmidt, M. (2003). Localized spectro-temporal features for automatic speech recognition. In *Proc. Eurospeech/Interspeech, Geneva*.

Kleinschmidt, M. and Gelbart, D. (2002). Improving word accuracy with Gabor feature extraction. In *Proc. ICSLP, Denver*.

Körding, K., König, P. and Klein, D. (2001). Learning of sparse auditory receptive fields. In *Proc. IJCNN*.

Lieb, M. and Fischer, A. (2001). Experiments with the philips continuous asr system on the aurora noisy digits database. In *Eurospeech*.

Lieb, M. and Fischer, A. (2002). Progress with the philips continuous asr system on the aurora noisy digits database. In *ICSLP*.

Lippmann, R. (1997). Speech recognition by machines and humans. *Speech Communication*, **22**:1–15.

Miller, L., Escabi, M., Read, H. and Schreiner, C. (2002). Spectrotemporal receptive fields in the lemniscal auditory cortex. *J. Neurophysiol.*, **87**:516–527.

Nadeu, C., Macho, D. and Hernando, J. (2001). Time and frequency filtering of filter-bank energies for robust HMM speech recognition. *Speech Communication*, **1–2**:93–114.

Qui, A., Schreiner, C. and Escabi, M. A. (2003). Gabor analysis of auditory midbrain receptive fields: Spectro-temporal and binaural composition. *Neurophysiology*, **90**:456—476.

Rabiner, L. and Juang, B.-H. (1993). *Fundamentals of Speech Recognition*. Prentice Hall, New York, revised edition.

Schreiner, C. and Calhoun, B. (1994). Spectral envelope coding in cat primary auditory cortex: properties of ripple transfer functions. *Auditory Neuroscience*, **1**:39–61.

Schreiner, C., Read, H. and Sutter, M. (2000). Modular organization of frequency integration in primary auditory cortex. *Annual Review Neuroscience*, **23**:501–529.

Schukat-Talamazzini, E. G. (1995). *Automatische Spracherkennung*, Vol. unknown of *unknown*. Vieweg Verlag, Braunschweig.

Somervuo, P., Chen, B. and Zhu, Q. (2004). Feature transformation and combinations for improving asr performance. *insert here*.

Van Compernolle, D. and Claes, T. (1996). Snr-normalization for robust speech recognition. In *ICASSP*.

Danksagungen

Zunächst möchte ich mich herzlich bei Prof. Dr. Dr. Birger Kollmeier dafür bedanken, dass er mir die Anfertigung dieser Arbeit ermöglicht und durch hilfreiche Anregungen ihren Fortgang unterstützt hat. Besonders herzlich danke ich Dr. Michael Kleinschmidt, von dessen großer Erfahrung ich profitieren konnte. Durch seine aufgeschlossene Art und seine große Kompetenz haben mir Diskussionen mit ihm viel Freude bereitet.
Ein großer Dank geht ebenfalls an Dr. Volker Hohmann für seine wertvollen Vorschläge, die in diese Arbeit eingeflossen sind. Heiko Gölzer danke ich für die vielen Tips und Diskussionen. Mögen immer viele Schälchen auf deinem Tablett stehen!
Ronny Meyer und Johannes Nix ist es zu verdanken, dass ich die Arbeit im Büro trotz lärmender Festplatten als angenehm empfunden habe.

Bei allen Mitgliedern der AG Medi bedanke ich mich für die angenehme Athmosphäre, die nicht nur in fachlichen Diskussionen spürbar war, sondern auch bei gemeinsamen Aktivitäten vom Sommerfest bis zu Volley- und Basketball.

Mein Dank geht insbesondere auch an Dr. Alexander Fischer, der mir viele Hilfestellungen und Anregungen in meiner Zeit bei Philips gegeben hat, und der nicht müde geworden ist, meine vielen Nachfragen in der Zeit danach zu beantworten. Auch Dr. Christoph Neukirch danke ich für viele Tips im Umgang mit Aspirin und das Setzen von Dateirechten. Für seine große Gastfreundschaft bin ich auch Leo Bosch zu Dank verpflichetet.

Meiner Familie und meiner Freundin Julia danke ich herzlich für die viele Unterstützung und für's "Rückenfreihalten".

Acknowledgements

I'm indepted to David Gelbart, who gave me countless tips and lots of advice, especially during my stay at Philips.

Hiermit versichere ich, daß ich diese Arbeit selbstständig verfaßt und keine anderen als die angegebenen Hilfsmittel und Quellen benutzt habe.