Quantile Based Histogram Equalization for Noise Robust Speech Recognition

Von der Fakultät für Mathematik, Informatik
und Naturwissenschaften
der Rheinisch-Westfälischen Technischen Hochschule Aachen
genehmigte Dissertation zur Erlangung des akademischen
Grades eines Doktors der Naturwissenschaften

von

Diplom–Physiker Florian Erich Hilger

aus

Bonn – Bad Godesberg

Berichter: Univ.-Prof. Dr.-Ing. Hermann Ney
Hon.-Prof. Dr. phil. nat. Harald Höge

Tag der mündlichen Prüfung: 6. Dezember 2004

Acknowledgements

I would like to thank my supervisor Prof. Dr.–Ing. Hermann Ney for his constant support, his valuable advice, and giving me the opportunity to realize this work at the Lehrstuhl für Informatik VI in Aachen.

Prof. Dr. phil. nat. Harald Höge from Siemens AG Munich kindly took over the role of the second supervisor. I would like to thank him for his interest in this work and his suggestions.

The joint work of my colleagues from the speech recognition group provided the necessary foundation, on which I could build my research. I would like to express my gratitude for the contributions of Maximilian Bisani, Stephan Kanthak, Klaus Macherey, Wolfgang Macherey, Sirko Molau, Michael Pitz, Ralf Schlüter, Achim Sixtus, Tibor Szilassy, Frank Wessel, and András Zolnay. I also owe a lot to numerous other fellow researchers and project partners, who gave me valuable feedback and advice.

Thanks to the people from the image processing and translation group for inspiring discussions about their research fields. The system administration team did a great job in providing a reliable environment for the experimental evaluations.

Special thanks to Andrea Semroch for her support and patience.

This work was partially funded by the DFG (Deutsche Forschungsgemeinschaft) under the contracts NE 572/4-1 and NE 572/4-3, and the European Commission under the Human Language Technologies project TRANSTYPE2, IST-2001-32091.

Abstract

This work describes an algorithm to increase the noise robustness of automatic speech recognition systems.

In many practical applications recognition systems have to work in adverse acoustic environment conditions. Distortions and noises caused by the transmission are typical for telephone applications. Considerable amounts of varying background noise are a problem for all mobile applications such as cellular phones or speech controlled systems in cars. Automatic systems are much more sensitive to the variabilities of the acoustic signal than humans. Whenever there is a mismatch between the distribution of the training data and the data that is to be recognized, the recognition word error rates will increase.

There are three possible ways of dealing with such a mismatch during the recognition process: adapting the recognizer's model parameters to the current noise condition, using a modified likelihood calculation that is invariant to the distortions caused by the noise, and reducing the influence of the noise during feature extraction. Within this work a feature extraction method is investigated.

The goal was to develop a computationally inexpensive method that can be applied in real time online systems. It should not require any prior assumptions about the noise conditions that are to be expected or the kind of training data available. And it should be independent from the actual recognition system that will use the features.

Quantile based histogram equalization improves the recognition performance in noisy conditions by applying a non–linear parametric transformation function during feature extraction. It reduces an eventual mismatch between the current recognition data and the distribution of the data the system was trained on. Based on the quantiles of the cumulative distributions, the parameters of the transformation function can be reliably estimated from small amounts of data. By the way they are defined quantiles are independent from the scaling, range, and amount of data. Thus making the method independent from prior assumptions about the training and recognition data.

The approach is integrated into a modified Mel cepstrum feature extraction, in which the logarithm is replaced by a root function to increase the noise robustness. The actual transformation that is proposed in this work consists of two steps. First, a power function transformation is applied to each output of the Mel–scaled filter–bank, then neighboring filter are channels combined linearly. These transformation steps can be added to the feature extraction using a moving window implementation that does not require more delay than a conventional mean and variance normalization.

To investigate the genericity of the approach and the proposed setup, experimental evaluations have been carried out with different speech recognition systems, on several databases with different levels of complexity, ranging from digit strings (SpeechDat Car) to larger vocabulary isolated word (Car Navigation) and continuous speech recognition tasks (Wall Street Journal with added noise).

Consistent recognition results were observed on all databases. The modified feature extraction, with the root instead of the logarithm, already outperformed the original baseline on noisy data. Filter channel specific quantile equalization always improved these results, yielding relative improvements between of 5% and 50%, depending on the recognition task and the mismatch of the data. Finally, the combination of neighboring filter channels was able to reduce the error rates somewhat further, especially if the noise, like car noise, was band limited.

Zusammenfassung

Diese Arbeit beschreibt einen Algorithmus zur Verbesserung der Geräuschrobustheit von automatischen Spracherkennungssystemen.

In vielen praktischen Anwendungen müssen Spracherkennungsysteme unter ungünstigen akustischen Umgebungsbedingungen arbeiten. Verzerrungen und Rauschen sind typisch für Anwendungen im Bereich der Telefonie. Erhebliche, wechselnde Hintergrundgeräusche sind ein Problem bei allen mobilen Anwendungen, wie Mobiltelefonen oder sprachgesteuerten Systemen in Fahrzeugen. Automatische Systeme reagieren viel empfindlicher als Menschen auf Variabilitäten im akustischen Signal. Sobald sich die Verteilung der Trainingsdaten von derjenigen der zu erkennenden Daten unterscheidet, steigen die Wortfehlerraten bei der Erkennung.

Es gibt drei prinzipielle Möglichkeiten, während der Erkennung mit einem solchen Unterschied umzugehen: Eine Anpassung der Modellparameter des Erkenners an die aktuellen Geräuschbedingungen, eine modifizierte Wahrscheinlichkeitsberechnung die invariant gegenüber den Veränderungen durch die Geräusche ist und eine Reduktion des Einflusses der Geräusche während der Merkmalsextraktion. Im Rahmen dieser Arbeit wird eine Methode im Merkmalsbereich untersucht.

Das Ziel war es, eine Methode zu entwickeln, die wenig Rechenaufwand erfordert und in Echtzeitsystemen eingesetzt werden kann. Sie soll keine a-priori Annahmen über die zu erwartenden Geräuschbedingungen oder das zur Verfügung stehende Trainingsmaterial erfordern. Und sie soll unabhängig vom letztlich eingesetzten Spracherkennungsystem sein.

Der auf Quantilen basierende Histogram–Ausgleich verbessert die Erkennung durch das Anwenden einer nichtlinearen parametrischen Transformationsfunktion. Sie reduziert einen etwaigen Unterschied zwischen den Verteilungen der Erkennungs– und Trainingsdaten. Basierend auf den Quantilen der kumulativen Verteilungen lassen sich die Parameter der Funktion zuverlässig auf kleinen Datenmengen schätzen. Per Definition sind die Quantile unabhängig von der Skalierung, dem Wertebereich und der Datenmenge. Damit ist die Methode unabhängig von Annahmen über Trainings– und Testdaten.

Das Verfahren wird in eine modifizierte Mel Cepstrum Merkmalsextraktion integriert, bei der anstelle des Logarithmus zur Verbesserung der Geräuschrobustheit eine Wurzelfunktion eingesetzt wird. Die eigentliche Transformation, die im Rahmen dieser Arbeit eingesetzt wird, besteht aus zwei Schritten. Zunächst wird eine Potenzfunktion auf die Ausgänge der Mel–skalierten Filterbank angewandt, danach werden benachbarte Filterkanäle linear kombiniert. Unter Verwendung eines laufenden Fensters können diese Transformationsschritte so in die Merkmalsextraktion integriert werden, dass dabei nicht mehr

Verzögerung als bei einer konventionellen Mittelwerts– und Varianznormierung erforderlich ist.

Um die Verallgemeinerbarkeit des Verfahrens zu untersuchen, wurden Experimente mit verschiedenen Spracherkennungssystemen auf unterschiedlichen Datensätzen durchgeführt: von Ziffernketten (SpeechDat Car) bis hin zu Erkennungsaufgaben mit einem größeren Vokabular von Einzelworten (Car Navigation) und kontinuierlicher Sprache (Wall Street Journal mit Geräuschen unterlegt).

Auf allen Datensätzen wurden konsistente Erkennungsergebnisse beobachtet. Die modifizierte Merkmalsextraktion, mit der Wurzelfunktion an Stelle des Logarithmus, lieferte auf verrauschten Daten bereits bessere Erkennungsergebnisse als das Original. Die auf Quantilen basierende Transformation individueller Filterkanäle konnte diese Ergebnisse immer verbessern, abhängig von der Erkennungsaufgabe und dem Missverhältnis zwischen den Trainings– und Testdaten lagen die relativen Verbesserungen zwischen 5% und 50%. Schließlich konnte die Kombination benachbarter Filterkanäle die Fehlerrate noch etwas weiter reduzieren, insbesondere bei bandbegrenzten Geräuschen wie beispielsweise Fahrgeräuschen in Autos.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Automatic speech recognizers are systems that process speech signals, in order to obtain the written word sequence that corresponds to the spoken utterance. Speech is the fundamental form of communication between humans, so a lot of convenient applications for automatic speech recognition systems can be thought of. However, the diversity and variability of speech signals make the development of automatic recognition systems a difficult and challenging task.

Compared to the very long history of transcribing speech as written language, the development of methods to automatize this task is recent. Sets of written symbols that represent the units of a language have already been developed several thousand years ago. In the second half of the nineteenth century an important step towards automatic processing of speech was taken, with the development of methods to record and transmit acoustic signals. The first attempts to actually recognize recorded phonemes and words have then been made in the 1940's and 50's (cf. historical review in [Rabiner and Junag 1993]). However, really significant progress has only been made in the last two decades, in which the developments in microelectronics have made the practical implementation of more sophisticated approaches to speech recognition possible. The systems have evolved from crude small vocabulary research prototypes to a wide variety of applications.

These applications can be divided into two board classes: actual transcription systems and systems that use speech as natural, intuitive interface.

- **Dictation and transcription systems:** given a user's spoken input or recorded speech data, these systems aim at putting out the correct literal transcription of what was said.

  Dictation systems are the typical speech recognition application most people think of first. Commercial office dictation systems for large vocabulary continuous speech are available for different tasks, e.g. in the legal domain or for medical reports. In more general sense, mobile systems to fill in forms, take notes or write short messages in a hands–free mode can also be considered a special case of dictation.

  Transcription systems aim at converting any kind of recorded speech data into written text, without directly interacting with the speaker. Systems that transcribe television or radio broadcasts are typical applications which are of interest for archiving and monitoring news for intelligence purposes.

- **Systems with speech interfaces:** these systems use speech as natural, intuitive interface. The goal is giving a satisfactory system response that corresponds to the wishes of the user in a minimal number of interaction steps. Reliable speech recognition is important for these systems too, but the correct literal transcription of everything the user said is not necessarily required.

  Small vocabulary command and control applications that recognize isolated words or predefined phrases can be applied whenever it is more convenient or safer to use speech as interface. Voice dialing is available in many mobile–phones on the market today. Hands free speech driven controls in a cars increase the safety, the driver is not distracted by looking for the right buttons to push.

  Spoken dialog systems can be used to make information available via telephone around the clock, without maintaining cost-intensive human operated call centers. Early dialog systems were based on small sets of words the users could say, to answer a more or less annoying fixed sequence of predefined questions. The recent development is towards more flexible mixed initiative systems. They allow the users to express their requests using natural sentences. To name but a few applications, dialog systems can be used for time–table information, hotel reservation, telephone directory assistance and user specific traffic or whether information services.

Despite all the improvements that have been made, the problem of automatic speech recognition can not be considered to be solved. The automatic systems that are available are still far from the human ability to reliably recognize speech with arbitrary content, even under adverse acoustic environment conditions. There still are no general all purpose speech recognition systems, even large vocabulary speech recognizers are not able to recognize utterances with arbitrary content, their domain of application is always limited.

The list of applications above also shows the need to develop systems that are not restricted to a quiet acoustic environment and a clearly defined recording setup. This adds an other dimension to the problem of automatic recognition. As soon as the speech that is to be recognized is not recorded in a quiet environment with a specific microphone, noise, distortions and changing channel characteristics come into the play and have to be dealt with.

If no specific techniques to increase the noise robustness are applied, even background noises that do not significantly affect the intelligibility for human listeners can make the recognition error rate of current automatic systems rise in an unacceptable extent.

Within this work a specific method to increase the noise robustness shall be investigated. After giving a short introduction to the statistical speech recognition approach in general [Jelinek 1997], some considerations about other techniques to increase the noise robustness will lead to the introduction of the quantile based histogram equalization method.

Like the speech recognition system itself, the quantile equalization method is based on a statistical framework. If noise or distortions cause a systematic mismatch between the distribution of the data that is to be recognized and distribution of the data the system was trained on, a transformation is applied to reduce the mismatch and thus increase the recognition performance.

Chapter 2

Automatic Speech Recognition

2.1 Introduction

The large variability of speech signals is a fundamental problem for automatic recognition systems. Even if the same speaker utters the same sentence several times, the loudness, the speaking rate and the intonation can differ significantly. The individual words can be articulated clearly or mumbled without distinct pauses between the words.

In addition to the variabilities that are caused by the individual speaker, speaker independent recognition systems have to cope with different voices, speaking styles and dialects. The characteristics of the voice are for example influenced by the particular anatomy of the vocal tract, the gender and the age of the speaker.

Once the speech signal leaves the mouth of the speaker it is exposed to various kinds of distortions. Background noises and cross talk from other speakers can interfere with the signal. The microphone characteristics and the amplifier of the recording device also influence the signal, as well as an eventual distortions due to coding or transmission after the recording. Figure 2.1 illustrates the different sources of variability and distortion that can influence the speech signal before it is actually processed by the recognizer.

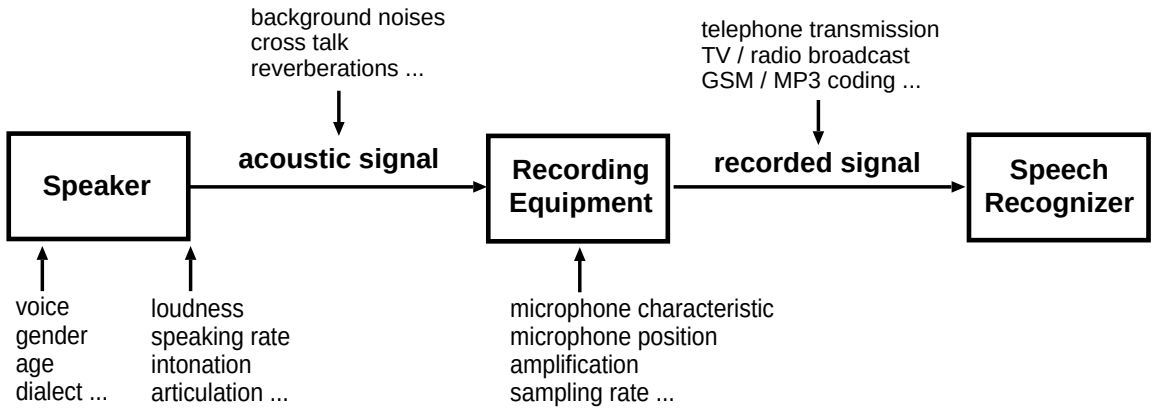

Figure 2.1: Sources of variability and distortion that can influence a speech signal.

These variabilities do not contain any information about the content of the utterance itself, so the speech recognition system has to ignore them. On the other hand, subtle variations of the signal can be important for the distinction between different sounds and words. In consequence, the goal of speech recognition research is to develop systems that can recognize a spoken word sequence by efficiently ignoring the variabilities that do not contribute to the recognition, while at the same time reliably extracting the relevant content from the signal.

Systems based on statistical methods have proven to be successful for that task and are now established as standard approach. These systems can make correct decisions based on uncertain data and vague knowledge that is learned automatically from training data.

2.2 Statistical Speech Recognition

In a statistical framework the goal of finding the written word sequence that corresponds to a spoken utterance can be expressed as follows:

Given the acoustic speech signal, the most probable word sequence has to be found. The acoustic signal is represented by a sequence of acoustic feature vectors $x_1^T = x_1, \ldots, x_T$ which are extracted from the original signal. The task of the statistical speech recognition system is to find $\{w_1^N\}_{opt} = \{w_1, \ldots, w_N\}_{opt}$, the optimal word sequence, which maximizes the conditional probability $p(w_1^N|x_1^T)$ given the sequence of feature vectors [Bahl et al. 1983]. Under the assumption that the true distribution is known, the word sequence that maximizes the posterior probability minimizes the probability of sentence errors, e.g. [Duda and Hart 1973]. Using Bayes decision rule [Bayes 1763] the maximization can be rewritten as follows [Bahl et al. 1983]:

$$\{w_1^N\}_{opt} = \underset{w_1^N}{\operatorname{argmax}} \left\{ p(w_1^N|x_1^T) \right\} \tag{2.1}$$

$$= \underset{w_1^N}{\operatorname{argmax}} \left\{ \frac{p(x_1^T|w_1^N) \cdot p(w_1^N)}{p(x_1^T)} \right\} \tag{2.2}$$

$$= \underset{w_1^N}{\operatorname{argmax}} \left\{ p(x_1^T|w_1^N) \cdot p(w_1^N) \right\} \tag{2.3}$$

The probability of the feature vectors $p(x_1^T)$ in equation 2.2 can be omitted, it is independent from the word sequence w_1^N and does not influence the maximization process.

The conditional probability of the feature vectors given a hypothesized word sequence $p(x_1^T|w_1^N)$ is the so called acoustic model probability. The prior probability of the written word sequence $p(w_1^N)$, independent from the acoustics is denoted language model probability. The unknown true probability distributions are approximated by probabilities that are estimated on training data using the model assumptions described in the following sections.

Figure 2.2 illustrates the four main components an automatic speech recognition system needs to evaluate equation 2.3 and find the most probable word sequence [Ney 1990]:

- **Feature Extraction:** based on short time spectral analysis a sequence of acoustic feature vectors x_1^T is extracted from the speech signal.

- **Acoustic Model:** to calculate $p(x_1^T|w_1^N)$ words are modeled as sequence of hidden Markov model states. In small vocabulary systems the whole word models are used. Medium and large vocabulary systems build up the models for the words from phoneme models that are concatenated according to a pronunciation lexicon.

- **Language Model:** the language model probability $p(w_1^N)$ is independent from the acoustic signal. Syntactical constraints, semantics and pragmatics of a language make certain written word sequences more probable than others are, the language model provides these probabilities.

- **Search:** the acoustic model and the language model are the knowledge sources of the speech recognition system. The search module combines them according to Bayes decision rule and determines the word sequence with the highest posterior probability $\{w_1^N\}_{opt}$.

The following sections will describe these fundamental system modules in more detail.

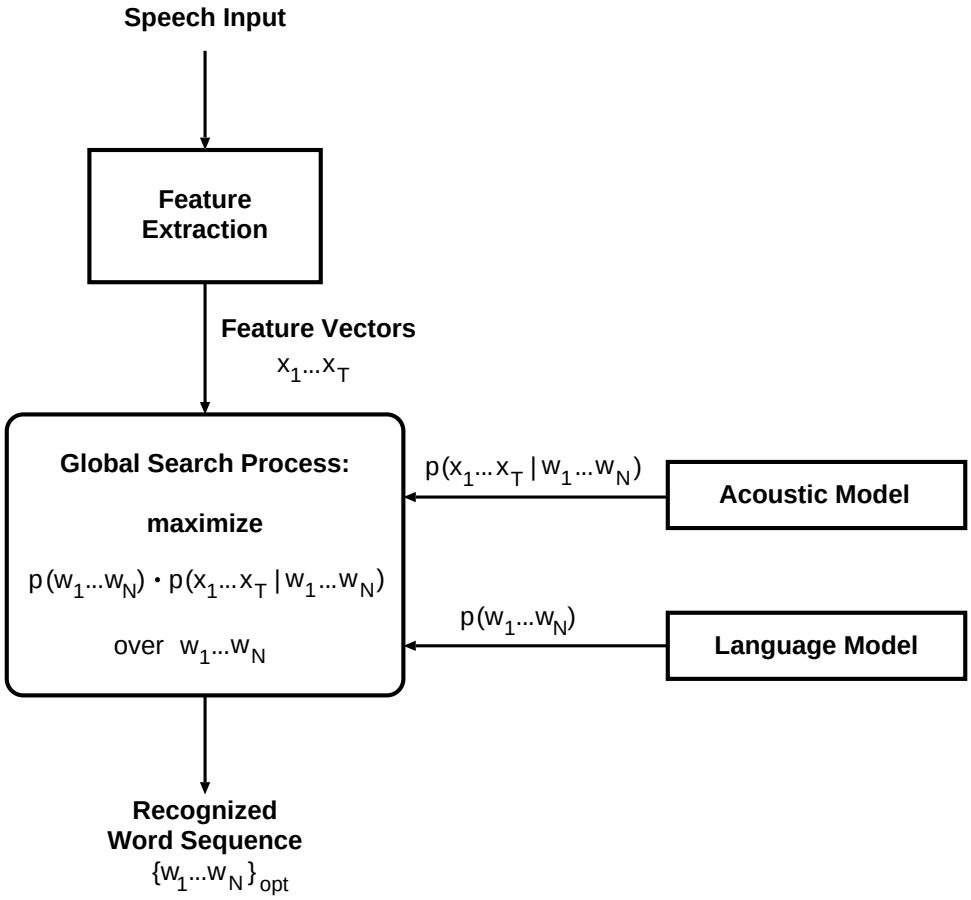

Figure 2.2: System architecture of a speech recognizer using Bayes decision rule.

2.3 Feature Extraction

The feature extraction is the process in which the raw samples of the speech signal are converted into the sequence of feature vectors x_1^T that are actually used for the recognition process [Rabiner and Schafer 1978].

Typically a feature vector is extracted every 10ms. The goal of the feature extraction is to provide feature vectors of low dimensionality that allow a good distinction between the spoken phonemes. At the same time these features should be invariant with respect to variabilities of the signal that do not influence the decision process. Eventual variabilities of the speaker, the transmission channel and background noise should have little impact on the features.

Within this work so called Mel–frequency cepstral coefficients (MFCC) will be used as features [Davis and Mermelstein 1980]. A schematic overview of the baseline feature MFCC extraction used in the RWTH speech recognition system is depicted is figure 2.3.

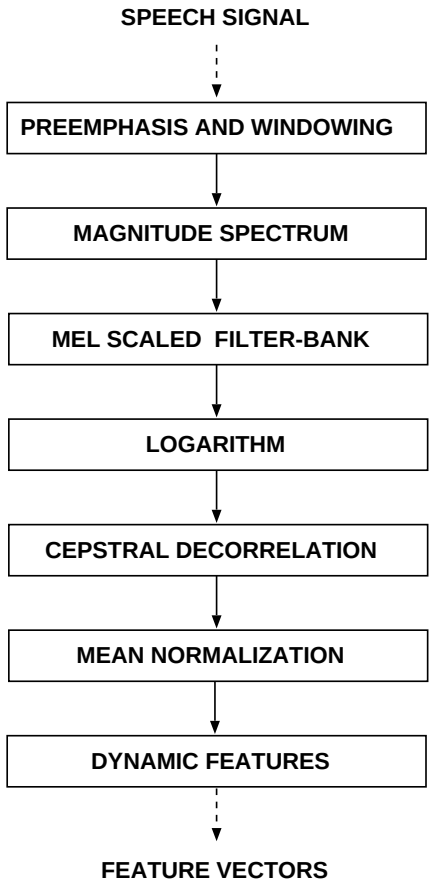

Figure 2.3: Baseline Mel–frequency cepstral coefficient feature extraction.

Preemphasis and Windowing

The raw speech signal is given as a sequence of sample values. The sampling rate depends on the frequency bandwidth of the signal. For telephone applications a sampling rate of

8kHz is sufficient, otherwise a sampling rate of 16kHz is typically used. In the first step of the actual feature extraction the signal is differentiated (preemphasis) by calculating the difference between succeeding samples s_n:

$$d_n = s_n - s_{n-1} \tag{2.4}$$

The spectral energy of speech signals usually decreases with increasing frequency. The differentiation corresponds to high pass filtering that emphasizes the spectral energy in the higher frequency range.

After the preemphasis the signal is blocked into overlapping segments of 25ms length, so called time frames, assuming that speech signals are stationary within such a window. The windowing function that defines the segments is shifted along the time axis is 10ms steps. The window could be any type of function that cuts a short segment from the signal. Usually the so called Hamming window function is used. If N_s is the number of samples in the window the Hamming window is defined as:

$$h_n = 0.54 - 0.46 \cos\left(\frac{2n\pi}{N_s - 1}\right) \tag{2.5}$$

Windowing in the temporal domain corresponds to a convolution with the frequency response of the window in the frequency domain [Rabiner and Schafer 1978]. The Hamming window has the advantage of having low side lobes in the frequency domain. Compared to other types of windows, e.g. a simple rectangular window, the frequency spectrum of the original speech signal will not be distorted that much if a Hamming window is applied.

Magnitude Spectrum

The magnitude spectrum of the windowed segments is calculated by applying a fast Fourier transform (FFT) [Cooley and Tuckey 1965], typically of length $N_{FFT} = 512$ or 1024. If the number of samples N_s in the window is lower than the FFT length N_{FFT}, a corresponding number of zeros is added (zero–padding).

For real valued input signals the resulting complex FFT coefficients are symmetric, so only $N_{FFT}/2$ coefficients are used in the following processing steps.

Mel–Scaled Filter–Bank

The frequency resolution of the human ear decreases towards higher frequencies. To model this effect the frequency axis f is warped to Mel–scale f_{Mel} [Stevens et al. 1937, Stevens and Volkmann 1940] by applying a logarithmic warping function:

$$f_{Mel} = 2595 \log_{10}\left(1 + \frac{f}{700\text{Hz}}\right) \tag{2.6}$$

After the frequency warping a filter–bank of bandpass filters is applied. The filters (15–25 depending on the frequency range that is considered) are overlapping

and equally spaced on the Mel–scaled frequency axis. A triangular weighting function (figure 2.4) determines the contribution of a frequency to the filter's output Y_k [Davis and Mermelstein 1980].

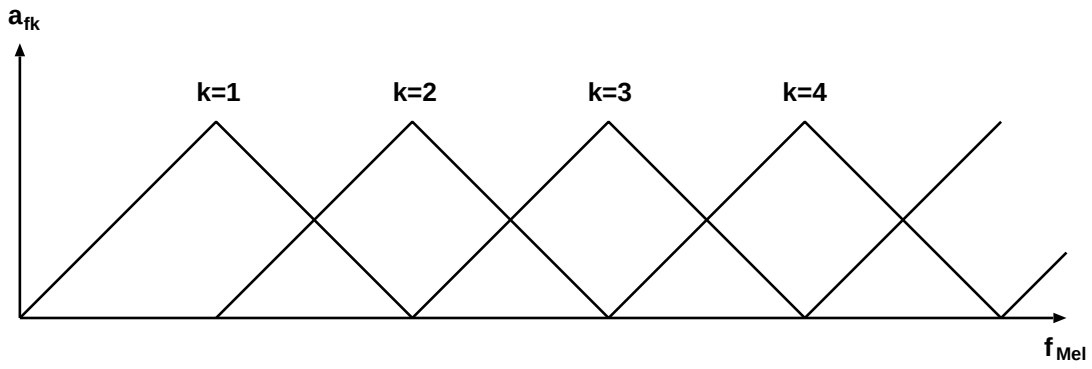

Figure 2.4: Overlapping filter–bank channels equally spaced on the Mel–frequency axis.

Logarithm

A logarithm is applied to the resulting output of the filter–bank to reduce the dynamic range of the signal. From a physiological point of view the logarithm mimics the non–linear dependency between intensity and loudness of a signal as it is perceived by humans. From the perspective of statistical pattern recognition, the variance for speech and silence portions is scaled to a similar range and the convolutional influence of a transmission channel is converted to a linear relation.

Cepstral Decorrelation

The overlap between the neighboring filters leads to a correlation between the filter channels which results in a covariance matrix that approximatively has a Toeplitz structure. Given the Toeplitz structure the cepstrum transformation [Bogert et al. 1963], a discrete cosine transform, can be used for the decorrelation [Davis and Mermelstein 1980]:

$$c_m = \sum_{k=1}^{K} Y_k \cos\left(\frac{\pi m(k-0.5)}{K}\right) \tag{2.7}$$

The resulting coefficients c_m are the Mel–frequency cepstral coefficients. The 0th coefficient corresponds to the logarithmic–energy of the current time frame. The highest cepstral coefficients, those that represent the details of the signal, are usually omitted. In the RWTH system 12 coefficients are used for data recorded over telephone lines with 8kHz sampling rate and 16 coefficients in the case of 16kHz sampling.

Mean Normalization

A simple but efficient normalization techniques that is used as standard in the RWTH feature extraction is mean normalization. It eliminates the influence of an unknown constant transmission channel by subtracting of the longterm mean from the cepstral coefficients or filter–bank outputs. More considerations about mean normalization can be found in the chapter on noise normalization techniques on page 24.

Dynamic Features

The temporal dynamics of the feature vectors can be taken into account by augmenting the cepstral feature vector with its derivatives, e.g. [Picone 1993]. The derivatives can be calculated as simple differences between succeeding time frames or by linear regression over some e.g. 5 consecutive time frames. In the baseline setup of the RWTH system linear regression is applied, the first derivative of all cepstral coefficients and the second derivative of the 0th cepstral coefficient are used. The resulting dimension of the actual feature vector then is $16 + 16 + 1 = 33$ in the case of 16kHz data.

Linear Discriminant Analysis

In the RWTH system linear discriminant analysis (LDA e.g. [Fukunaga 1990]) is used as final feature extraction step. Linear discriminant analysis is based on the idea of applying a matrix multiplication that reduces the dimensionality of the feature space while at the same time maximizing the separability of the phonetic classes.

Typically three succeeding feature vectors, including the static cepstral coefficients and their derivatives, are concatenated [Welling et al. 1997]. This vector is multiplied with the LDA matrix. After the multiplication the dimensionality is reduced to the original dimensionality. Alternatively seven succeeding vectors of static cepstral vectors can be concatenated and used as input vector for the LDA (cf. system descriptions in [Sixtus 2003]).

In the case of 16kHz speech data the resulting feature vector x that is used for recognition has 33 components. The complete sequence of feature vectors for the utterance that consists of 1 to T time frames is denoted x_1^T.

2.4 Acoustic Modelling

In Bayes decision rule (equation 2.3) the conditional probability $p(x_1^T|w_1^T)$ is required. It is the probability of observing the feature vector sequence x_1^T given an hypothesized word sequence w_1^N.

The size of the vocabulary needed for the recognition task and the amount of training data available determines how the words w are modeled. For small vocabulary command and control applications or digit recognition tasks, so called whole word models can be used. If the recognizer's vocabulary is increased, it is likely that even in an abundant amount of training data many words only occur rarely or not at all. The whole word model approach can not be used in these cases. Instead, subword models like syllables or phonemes have to be used to build up the models for words.

The phonemes are the smallest, theoretical units of sounds that can change the meaning of a words. Depending on the language the phoneme inventory consists of 40–50 units. Pronunciation lexica provide the phoneme sequence that corresponds to a word. The measurable acoustic realization of the phonemes depends on their context, the preceding and succeeding phonemes. In the RWTH speech recognition system the phonemes are modeled in a triphone context, were each phoneme is only dependent on its direct predecessor and successor [Ney 1990]. These context dependent phonemes, so called "triphones," are modeled as a sequence of hidden Markov models states s [Baker 1975, Rabiner 1989].

Hidden Markov models are stochastic finite state automata that define a set of states and the possible transitions between these. For each state s an appropriate probability distribution function defines the probability of emitting a feature vector x while being in the state. The typical variations of the speaking rate are modeled by the transitions between the individual states, transition probabilities are attached to each of these transitions.

The triphones are usually assumed to have three parts, beginning, middle and end. In the RWTH system each part is modeled as pair of identical states (BB, MM, EE). The example in figure 2.5 illustrates the states corresponding to the triphone $_{\text{z}}$ a:$_{\text{m}}$ from the word "example" with the phoneme sequence I g' z a: m p l. There are three possible transitions from each state : staying in the same state, moving to the next state and skipping one state (Bakis topology [Bakis 1976]).

The acoustic model probability $p(x_1^T|w_1^T)$ in the Bayes decision rule can be evaluated by using these stochastic automata. A network of states corresponding to an hypothesized utterance w_1^T can be defined by concatenating the hidden Markov models of the triphones to words and the words to a sentence (figure 2.5). Let s_1^T be a path through this network, then the probability of this path is $p(x_1^T, s_1^T|w_1^T)$. By summing over all possible paths the acoustic model probability can be calculated:

$$p(x_1^T|w_1^T) = \sum_{s_1^T} p(x_1^T, s_1^T|w_1^T) \tag{2.8}$$

$$= \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|x_1^{t-1}, s_1^t, w_1^N) \cdot p(s_t|x_1^{t-1}, s_1^{t-1}, w_1^N) \tag{2.9}$$

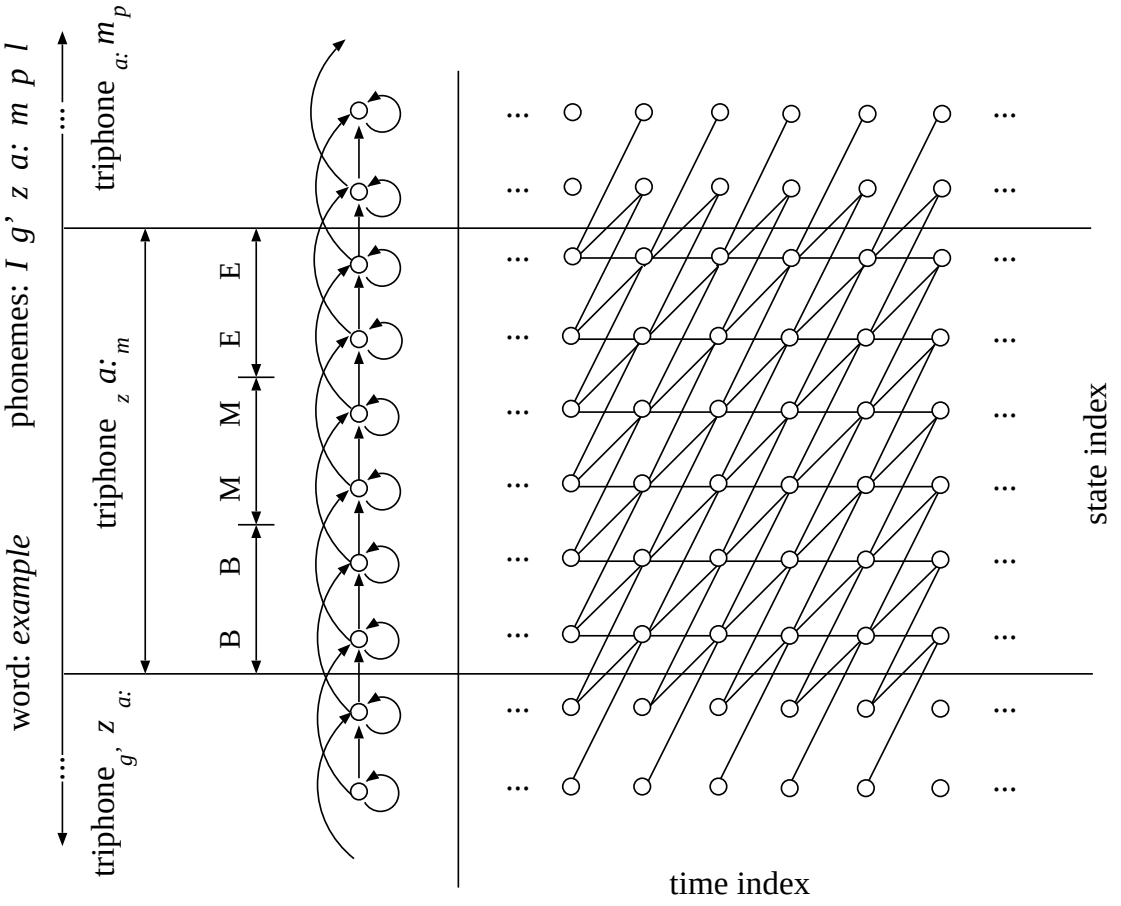

Figure 2.5: Hidden Markov Model with six states per triphone and transitions in Bakis topology.

This expression can be simplified under the assumption that the sequence of feature vectors is generated by a first order Markov process [van Kampen 1992], in which the probability of an acoustic observation does not depend on preceding observations:

$$p(x_1^T|w_1^T) = \sum_{s_1^T} \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \tag{2.10}$$

$$\approx \max_{s_1^T} \left\{ \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \right\} \tag{2.11}$$

Since the state sequence is given s_1^T in the equations, the dependence on the word sequence w_1^N can be omitted, it is redundant. To evaluate the sum over all the possible state sequences in equation 2.10 efficiently, a forward–backward algorithm [Baum and Petrie 1966] can be used. Alternatively, the full sum can also be approximated by only considering the most likely state sequence, the path with the maximal probability (equation 2.11). This so called Viterbi approximation allows an efficient time synchronous evaluation of the expression using dynamic programming [Bellman 1957, Viterbi 1967, Ney 1984, Ney 1990].

To model the emission probabilities $p(x_t|s_t)$ various methods have been proposed, using discrete probability distributions [Jelinek 1976, Liporace 1982], semi–continuous distributions [Huang and Jack 1989, Huang et al. 1990] or continuous distributions [Levinson et al. 1983, Ney and Noll 1988].

Within this work multimodal Gaussian distributions (Gaussian mixture densities) are used as representation of the continuous probability distributions. The emission probability for a state s is the weighted sum over the individual Gaussian densities. In the equation the densities are labeled with l with weighted c_{sl}. $\mathcal{N}$ denotes a Gaussian distribution with a mean μ_{sl} and covariance matrix Σ_{sl}.

$$p(x_t|s_t, w_1^N) = \sum_{l=1}^{L} c_{sl}\, \mathcal{N}(x_t|\mu_{sl}, \Sigma_{sl}) \tag{2.12}$$

$$\approx \max_{l} \{c_{sl}\, \mathcal{N}(x_t|\mu_{sl}, \Sigma)\} \tag{2.13}$$

In the practical implementation within the RWTH system equation 2.12 is simplified, equation 2.13. The sum over all densities is replaced by the maximum and the covariance matrix Σ_{sl} is replaced by a state and density independent pooled diagonal matrix Σ that can be estimated more reliably.

The densities' weights, means and variances for each phoneme state are the acoustic model parameters, they have to be estimated from training data, assuming that the statistics of the training data correspond to those of the data that is to be recognized. If this is not the case, the mismatch has to be reduced by special methods that are explained in chapter 3.

In most systems the training is carried out using a maximum likelihood framework. The expectation maximization algorithm [Dempster et al. 1977] is applied to iteratively optimize the parameters. In the RWTH system the Viterbi approximation (equation 2.11) is used [Viterbi 1967], so only the most likely state sequence contributes to the estimation of the model parameters.

In large vocabulary applications the problem arises that many context dependent phonemes only occur a few times or not at all in the training data. In order to obtain model parameters for them, without resorting to context independent monophone models, similar phonemes or phoneme states can be tied [Young 1992]. In the RWTH system a top down clustering algorithm is used to cluster the context triphones. The clustering is based on a phonetic classification and regression tree (CART) [Hwang et al. 1993, Young et al. 1994, Beulen et al. 1997].

The context dependencies between phonemes are not limited to the phonemes within words. Co–articulation over word boundaries occurs if the speaker does not make any distinct pauses between the words. The initial phoneme of a word is then influenced by the final phoneme of the predecessor word and vice versa. Taking into account these across–word dependencies [Hon and Lee 1991, Odell et al. 1994] improves the recognition performance at the cost of a significantly increasing computational complexity in the search process (section 2.6). In [Sixtus 2003] a detailed description of the efficient across–word model implementation in the RWTH system is given. Within this work across–word modelling was always used for the large vocabulary continuous speech recognition experiments.

2.5 Language Modelling

Independent from the acoustic speech signal the syntactical constraints, semantics and pragmatics of a language make certain word sequences more probable than others. The language model shall provide the prior probability $p(w_1^N)$ for a hypothesized word sequence w_1^N. Without having to code grammatical rules and constraints explicitly, the language model probabilities can be estimated form large collections of written text.

Usually the assumption is made that the word sequence can be modeled as Markov process in which the probability of a word w_n only depends on the predecessor words w_1^{n-1}:

$$p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \tag{2.14}$$

$$\approx \prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \tag{2.15}$$

The conditional probability for a word w_n given the so called history w_1^{n-1} can not be estimated for an arbitrary number of preceding words, equation 2.14. In practice, so called m–gram language models [Bahl et al. 1983] are used. Only m predecessor words are taken into account when calculating the word's probability (equation 2.15).

To estimate the conditional probabilities of the m–grams the maximum likelihood approach is used again. The evaluation criterion that is applied in the context of language modelling is the perplexity (PP) [Bahl et al. 1983]. It is defined as the inverse of the geometric mean of the language model probability for all words w_n in a sequence w_1^N:

$$PP = \left[\prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1})\right]^{-1/N} \tag{2.16}$$

This quantity can be seen as measure for the average number of possible words the recognizer has to choose from at each word position. The expectation is that the word error rate of the recognition process decreases if the perplexity is reduced.

If minimal perplexity is chosen to be the training criterion, a closed form solution for the estimation of the language model probabilities can be derived. The probabilities $p(w_n|w_{n-m+1}^{n-1})$ can be calculated by simply determining the relative frequency of the of the corresponding m–grams in a training dataset.

The problem with the relative frequency approach is the number of possible m–grams. It increases exponentially with m. Even when restricting the history to a length to 1 or 2, estimating so called bi– or tri–gram language models, many of these word pairs or triples will not occur in the training data. This problem can not be solved by simply increasing the amount of training data.

In order to assign probabilities larger than zero to the unseen m–grams smoothing methods have to be applied. These methods are based on discounting methods [Katz 1987, Ney et al. 1994, Generet et al. 1995, Ney et al. 1997]. Probability mass is

subtracted from some or all trigrams observed in training. This discounted probability mass is then distributed among the unseen (backing–off) or all (interpolation) m–grams. The specific amount of probability mass that is redistributed is based on a general language model probability distribution with a shorter history. The parameters of this generalized language model and the distribution parameters can be estimated automatically using a leaving–one–out approach, which is a special case of the cross–validation scheme [Ney et al. 1994]. A comparison of different smoothing techniques can be found in [Martin et al. 1999].

Many improvements of the baseline m–gram approach have been suggested. A language model cache [Kuhn and De Mori 1990, Generet et al. 1995, Martin et al. 1997] that stores the last few hundred recognized words can be used to adapt the language model probabilities to the topic of the current utterances or the specific vocabulary used by the speaker. Frequently occurring word sequences can be modeled by considering them as a phrase that is treated as one word [Jelinek 1991, Klakow 1998]. The m–grams that take into account the direct predecessors can be combined with distant m-grams that have gaps between the words [Rosenfeld 1994, Martin et al. 1999]. Word classes like proper names, locations, companies and date expressions can be used instead of distinct individual words [Brown et al. 1992, Kneser and Ney 1993, Jardino 1996, Martin et al. 1998]. Especially in the context of telephone directory assistance applications e.g. [Macherey and Ney 2003] this approach is beneficial.

The details of implementation of language models in the RWTH system are described in [Wessel et al. 1997].

2.6 Search

The search module determines the most probable word sequence $\{w_1^N\}_{opt}$ for a given sequence of acoustic feature vectors. The acoustic model and the language model are the knowledge sources that are used to calculate the posterior probability, with the expressions from equation 2.11 and equation 2.15 Bayes decision rule (equation 2.3) becomes:

$$\{w_1^N\}_{opt} = \underset{w_1^N}{\operatorname{argmax}} \left\{ \max_{s_1^T} \left\{ \prod_{t=1}^{T} p(x_t|s_t, w_1^N) \cdot p(s_t|s_{t-1}, w_1^N) \right\} \cdot \prod_{n=1}^{N} p(w_n|w_{n-m+1}^{n-1}) \right\} \quad (2.17)$$

The problem that makes the practical evaluation of this expression difficult is the number of possible word sequences. It increases exponentially with the maximal number of words N in the utterance. Compared to a naive implementation that evaluates all hypothesized word sequences independently, the complexity of the problem can be reduced significantly by applying dynamic programming [Bellman 1957]. The problem of finding the global optimal is decomposed into a sequence of successive local optimizations. Common partial results that are the same the initial part of different word sequences only have to be calculated once.

Two types of search algorithms based on dynamic programming are used in most speech recognition systems. Time–synchronous search using the Viterbi approximation [Viterbi 1967, Ney 1984, Ney 1990] and A* search, also known as stack decoding. In A* search the hypothesized hidden Markov model state sequences are expanded and searched, in way that is not time synchronous. The known probability of a state sequence that has already been evaluated is combined with an over–optimistic estimate of the probability of the unknown part of the state sequence that is to come [Jelinek 1969, Paul 1991]. Based on these probabilities the most likely hypothesis is expanded first. The result of the A* search largely depends on the heuristic estimate of the unknown probabilities.

In time–synchronous Viterbi search, the sequences of state hypotheses are evaluated and expanded on a synchronous time frame by time frame basis [Vintsyuk 1971, Baker 1975, Sakoe 1979, Ney 1984]. The probabilities of the different hypotheses can be compared directly at each time frame. Unlikely hypotheses can be eliminated to reduce the search space.

In the RWTH system the pronunciation lexicon is organized as a prefix tree of the phoneme sequences. This approach reduces the redundancy of the lexicon and the search space. Common prefixes that are part of different words in the lexicon only have to be evaluated once [Ney et al. 1992, Ney 1993, Ortmanns et al. 1997c] this reduces the search space significantly. However, in large vocabulary systems exploring the whole search space is not possible any more, even if this is done efficiently taking into account the redundancies.

In this case beam search has to be applied, unlikely hypotheses have to be eliminated by applying pruning methods [Ney et al. 1987, Ortmanns and Ney 1995]. When pruning unlikely hypotheses it can not be guaranteed that the globally best word sequence is found in the end. The word sequence that will be most likely at the end of the utterance can

be eliminated if it is temporarily probable than a competing hypothesis at a certain time frame. But these eventual search errors are no problem if the pruning parameters are adjusted appropriately. Usually the computational complexity can be reduced significantly by the pruning without compromising the recognition results.

Look–ahead techniques can make the pruning even more efficient. When organizing the lexicon as prefix tree the actual word identities are not known until the word end is hypothesized. At that point the language model probabilities can be taken into account. The language model look–ahead technique allows tighter pruning by considering the language model probabilities earlier. Even if the word identities are not known until the end of the words are reached, the number of possible ending words that can be reached from a certain state is limited. Among these the one with the highest language model probability can be determined. Its probability is an upper bound of the probability that can be expected in the best case, if it is low the branch of the tree can already be pruned without having to wait for the actual word ends. In addition to the language model look ahead, simplified acoustic models can be used to roughly estimate the acoustic model probability for the next few time frames. This phoneme look–ahead [Ney et al. 1992] method can eliminate unlikely hypotheses before spending computation time on calculating the correct acoustic model probabilities.

In large vocabulary speech recognition systems the number of Gaussian densities in the acoustic model is large, typically in the order of 10^4 to 10^5. The calculation of the emission probabilities requires the mayor part of the total computation time the speech recognizer needs, so methods that make the likelihood calculation faster can contribute to a significant reduction of the computational requirements. The methods that have been suggested for fast likelihood are usually based on vector quantization [Bocchieri 1993, Ortmanns et al. 1997b] or partitioning [Nene and Nayar 1996, Ortmanns et al. 1997b] of the feature space. The likelihood calculations can also be parallelized by using SIMD (single instruction multiple data) instructions [Kanthak et al. 2000a]. This method was used in the experiments with the RWTH system that are described in this work.

When the integration of complex acoustic and language models in an integrated single–pass search is difficult or computationally expensive, a multi–pass search framework can be used. The idea is to use simple models in a first recognition pass. The output of this preliminary recognition pass is a set of competing likely hypotheses. Among these, a second processing pass with more complex models can determine the single most likely one. The set of likely likely word sequences can be represented in different ways.

N–best lists contain the N word sequences with the highest posterior probabilities [Schwartz and Chow 1990]. The disadvantage of the N–best lists is the redundancy. Many hypothesized sentences only differ at a few word positions, the remaining identical word sequences are eventually stored several times. Word graphs provide a more compact representation of a significantly larger number of alternatives. The word hypotheses can be stored in a directed, acyclic weighted graph, in which the nodes represent word boundaries, i.e. starting and ending times of the hypothesized words, and the arcs represent the words themselves. Subsequences of identical words with identical boundaries are only stored once in the graph. Details on the generation of word graphs can be found in [Schwartz and Austin 1991, Ney et al. 1994, Aubert and Ney 1995, Woodland et al. 1995, Ortmanns et al. 1997a].

Obviously, multi–pass approaches can only be used in applications were a minimal error rate has larger priority than a real–time response of the system. The focus of this work is a single pass feature extraction method that can be applied in real–time online applications, so correspondingly only a single pass beam search approach was used. Multi–pass methods were not considered.

Chapter 3

Noise Robust Speech Recognition

3.1 Introduction

In many practical applications automatic speech recognition systems have to cope with noisy signals from adverse acoustic environment conditions. Distortions and noises caused by the transmission are typical for all kinds of telephone applications. In addition to these transmission channel distortions, considerable amounts of variable background noise are a severe problem for mobile applications, like cellular phones or speech controlled devices cars. Broadcast news transcription systems also have to deal with speech data that was not recorded in a quiet studio environment. Recordings made at press conferences, interviews and reports by field correspondents are usually affected by background noise.

Automatic systems are much more sensitive to these variabilities of the acoustic signal than humans. The recognition error rates of speech recognition systems that just use the standard methods described in the previous chapter will usually rise considerably in noisy conditions.

Up to a certain extent the noise robustness of the systems can be improved by simply providing training data that covers a broad range of noise conditions and channel characteristics. The examples shown in table 3.1 are the baseline results for the evaluations on the Aurora 4 database ([Hirsch 2002] and appendix on page 128). The database consists of read sentences from the Wall Street Journal that are corrupted by different artificially added noises at various signal-to-noise ratios and in some cases a microphone mismatch. For the example shown here, the standardized setup of the reference recognizer as described in [Parihar and Picone 2002] was used, together with a baseline MFCC feature extraction that did not contain any normalization or noise reduction [Hirsch and Pearce 2000].

The average error rate on the noisy data is significantly reduced from 69.8% to 39.6% by providing multicondition training data, but this is no satisfying solution to the problem of noise robust speech recognition: the resulting error rate is still much higher than the original error rate in matched clean conditions. Even abundant multicondition training data will not be able to cover all possible noise conditions and signal-to-noise ratios, so when using a system trained on noisy data, a systematic mismatch between the trained models of the recognizer and the data that is to be recognized can not be ruled out. The

Table 3.1: Recognition results on the Aurora 4 noisy WSJ database [Hirsch 2002] using the standardized baseline recognizer setup [Parihar and Picone 2002] and an MFCC feature extraction without any normalization. The result on the clean test data is compared to the average over the different added noise conditions and microphone channels.

	Word Error Rates [%]	
	clean data	noisy data
training data	test set 1	average test sets 1–14
original clean	14.9	69.8
noisy multicondition	23.5	39.6

recognition performance can still be poor. The increase of the error rate on clean data (from 14.0% to 23.5% in the example) is problematic too, especially shall be used in clean and noisy environments alike.

The real key to robust speech recognition is an improved feature extraction and the handling of the mismatch caused by the noise within the recognition system. Noise leads to undesired variabilities of the signal that have to be dealt with by the system. All variabilities that do not contribute to the actual classification task should be suppressed or ignored by the recognizer. The focus has to be put on the relevant content of the signal.

During the recognition process the emission probabilities for the feature vectors are calculated. Given a hypothesized hidden Markov model (HMM) state s and a feature vector x, the vector's emission probability is $p(x|s)$. The expression $p(x|s)$ suggests that there are three possible ways of dealing with a potential mismatch:

1. In the feature domain x the influence of the noise on the signal can be explicitly suppressed or removed using a noise estimate. Alternatively, the feature space can be considered as a whole and a transformation that inverses the changes caused by the noise can be applied. As a third alternative features that are inherently noise robust, i.e. not influenced significantly by the noise, can be extracted.

2. In the model domain s by using methods can be used to adapt the model parameters to the current noise condition. This can be done by approximating the influence of the noise on the model space and then applying a corresponding transformation or determining the difference between the current feature vectors and the models in a first recognition pass to estimate e.g. a transformation matrix that minimizes these differences. Combining clean speech and noise models to get appropriate models for the noisy speech is an other alternative.

3. During likelihood calculation $p(\cdot|\cdot)$ the noise robustness can be increased by either ignoring unreliable feature vector components, or using a distance calculation that is invariant to undesired variabilities.

Various approaches to noise robust speech recognition in these three domains have been investigated by many different research groups. The following three sections will give a review of some of these techniques. These considerations will then serve as motivation for the quantile based histogram equalization method.

3.2 Noise Robust Feature Extraction

A wide variety of feature domain approaches to noise robustness are based on the idea of explicitly removing the influence of the noise from the signal to enhance the speech, in order to obtain an estimate of the underlying original clean speech signal or feature vectors.

Spectral Subtraction

Spectral subtraction is a commonly used straight forward method to remove additive noise from speech signals. It was originally used for speech enhancement applications to increase the intelligibility of resynthesized speech for human listeners e.g. [Weiss et al. 1974]. The application in speech recognition feature extraction was suggested in [Boll 1979].

An estimate of the noise spectrum is subtracted from the signal in the magnitude, power spectrum or filter–bank domain. Assuming that the noise is additive, the current X_{ft} speech signal in frequency band f at time frame t is the sum of the original speech signal S_{ft} and the noise N_{ft}:

$$X_{ft} = S_{ft} + N_{ft} \tag{3.1}$$

Given an estimate of the noise $\bar{N}_{ft}$ a generalized spectral subtraction equation that provides the clean signal's estimate $\hat{X}_{ft} \approx S_{ft}$ can be written as:

$$\hat{X}_{ft} = \max\{\, X_{ft} - \alpha_{ft}\bar{N}_{ft}\,,\, \beta_{ft}\bar{N}_{ft}\,\} \tag{3.2}$$

α_{ft} is an overestimation factor. If the current noise at time frame t is larger than the average noise estimate that is subtracted peaks of so called "musical" noise remain in the signal. These peaks noise can be suppressed if an overestimation factor $\alpha_{ft} > 1$ is used [Berouti et al. 1979]. This factor can be a simple constant or a more complex function depending on the frequency and the current characteristics of the signal. In [Le Floc'h et al. 1992] the factor was made dependent on the average SNR of the frequency band. Considering the SNR at the current time frame t was suggested in [Lockwood and Boudy 1992]. The factor was increased in the low SNR portions of the signal, while the peaks of the speech parts were not distorted that much. In [Korkmazskiy et al. 2000] a speech silence detection was used to switch between different overestimation factors and a similar approach was used in [Hilger and Ney 2000], were the overestimation factor was a function of the noise estimate's variance and the subtraction was completely switched off during the speech portions of the signal.

Together with the overestimation factor a flooring factor or clipping limit β_{ft} has to be introduced to ensure that the values after the subtraction, to which a the logarithm will be applied, are positive. Like α_{ft} the factor β_{ft} can also be constant or an appropriate function of the signal.

The efficiency of spectral subtraction also largely depends on the estimation of the average noise, which requires a reliable speech silence detection or segmentation method. To avoid this decision problem a continuous estimation and subtraction method that only requires occasional updates of a noise HMM was proposed in

[Nolazco-Flores and Young 1994]. Alternatively the overall distribution of the data can be considered for the estimation. The use of common values in the histogram of the data was proposed in [Hirsch 1993]. In [Stahl et al. 2000] a quantile (e.g. the median) of the data's values was used as noise estimate.

In conclusion these considerations show that although the principal idea behind spectral subtraction is straight forward and simple, the efficient implementation requires a lot of optimizations that involve numerous parameters that have to be optimized empirically.

Wiener Filtering

Like spectral subtraction Wiener filtering [Wiener 1949] is a commonly used signal processing method that has found a wide range of applications domains like signal coding, signal restoration and channel equalization. It can be used whenever signal denoising is required, without being limited to speech processing applications.

Again under the assumption that the noise is additive (equation 3.1) a gain function W_{ft} is multiplied with the signal X_{ft} to obtain the clean speech estimate $\hat{X}_{ft}$:

$$\hat{X}_{ft} = W_{ft} X_{ft} \tag{3.3}$$

With given long term estimates of the power spectra for speech $\bar{S}_{ft}$ and noise $\bar{N}_{ft}$ a least means square error estimation of W_{ft} that minimizes the error between current estimate of the clean signal $\hat{X}_{ft}$ and the real signal S_{ft} yields the result:

$$W_{ft} = \frac{\bar{S}_{ft}}{\bar{S}_{ft} + \bar{N}_{ft}} \tag{3.4}$$

The estimates for the speech $\bar{S}_{ft}$ and noise $\bar{N}_{ft}$ also require a reliable speech silence detection.

Various implementations of this method for speech recognition purposes have been studied for a long time e.g. [Lim and Oppenheim 1978, Bernstein and Shallom 1991, Andrassy et al. 2001]. The efficiency of Wiener filtering was also shown within the ETSI Aurora evaluations [Pearce 2000]. The goal of these evaluations is to develop a standard method for a noise robust feature extraction that can be used in mobile applications. The advanced feature extraction that was standardized includes a two stage Wiener filtering [Macho et al. 2002, ETSI 2002].

Vector Taylor Series

The vector Taylor series method [Moreno et al. 1996] goes further. In addition to the noise estimate it uses a clean speech model and a its relation to noisy speech given by a Taylor series expansion to get the estimate of the clean speech vectors.

In the logarithmic spectral domain the effect of an additive noise N_{ft} and a multiplicative gain Q_{ft} can be expressed as:

$$X_{ft} = S_{ft} + Q_{ft} + \log(1 + \exp(N_{ft} - S_{ft})) \tag{3.5}$$

$$= S_{ft} + f(N_{ft}, S_{ft}, Q_{ft}) \tag{3.6}$$

Here X_{ft}, N_{ft} and S_{ft} are the noisy speech signal, the noise and the unknown original speech respectively in the logarithmic domain. Using this expression the clean speech estimate $\hat{X}_{ft}$ can be written as:

$$\hat{X}_{ft} = \int (X_{ft} - f(N_{ft}, S_{ft}, Q_{ft}))\, p(S_{ft}|X_{ft})\, dX_{ft} \tag{3.7}$$

This expression can be evaluated given the Taylor series expansion of f that relates the clean and noisy speech, a clean speech model estimated beforehand on clean training data, and a sample of the current noise to estimate the distribution of the noise [Moreno 1996].

Instead of the 1st order Taylor series approximation of f that is described in [Moreno et al. 1996] a polynomial approximation was suggested in [Raj et al. 1997]. The approach can also be modified to make it an adaptation method that provides the transformation in the model domain.

The estimation of the noise is again an important issue that requires special consideration, especially in non–stationary environment conditions. In [Kim 1998] a sequential estimation of the noise that can account for non–stationary environment conditions is suggested.

These and other speech enhancement methods based on prior assumption that the noise is additive and/or convolutional have proven their capabilities in many investigations, but from a theoretical point of view there is room for improvement. The assumption that the distortions are additive and/or convolutional is certainly a good starting point, but recordings made in real noisy environment are different from recordings with artificially added noises. The speakers will be influenced by the environment, triggering the so called Lombard effect e.g. [Junqua 1993, Junqua et al. 1999]. A person that is confronted with a noisy environment will start speaking different way, especially louder and with a different articulation. This increases the intelligibility for human listeners, but compromises automatic recognition. The effect is highly non–linear and difficult to model, so it would be better to have an approach that works without any prior assumptions about how the signal is distorted.

Explicitly suppressing the noise is only one way of making the extracted features robust in changing acoustic conditions. Suppressing the noise and enhancing the speech is important, but not the crucial point. What really matters is invariance. The features that are extracted should be are invariant to all variations in the signal that do not contribute to the distinction of classes during recognition process. As long as a systematic mismatch between the model distribution and the data that is to be classified remains, a statistical classifier will not reach its minimal classification error rate.

As alternative to noise removal or suppression the invariance can also be obtained by and reducing an eventual mismatch directly on the level of the data's probability distributions. This consideration leads to more general methods that are based on the statistics of the incoming data as a whole, they have the advantage that no prior assumptions about the nature noise and how it influences the signal have to be made.

Cepstral mean and variance normalization

The mean, the first moment of the probability distribution, should be dealt with first. Cepstral mean normalization (CMN) is a very simple but efficient method to reduce transmission channel characteristics. Originally used for speaker identification applications e.g. [Atal 1974, Furui 1981, Rosenberg et al. 1994] it has become a standard method in speech recognition. The longterm mean $\bar{x}_{ct}$ is calculated utterance wise or within a moving window e.g. [Viikki and Laurila 1998] making it dependent of the time t. No distinction between speech and silence portions of the signal is made. Then the transformed cepstral feature vector $\hat{x}_{ct}$ is then calculated by simply subtracting the mean from the current value x_{ct}:

$$\hat{x}_{ct} = x_{ct} - \bar{x}_{ct} \tag{3.8}$$

This subtraction in the logarithmic cepstral domain corresponds to the removal of a constant gain caused by the transmission channel in the spectral domain. Independent from this view, it can also be considered as first step in the direction of transforming vectors in a general feature space to reduce the mismatch between distributions form different environment conditions. Since the cepstrum transformation is just a linear operation the mean normalization can alternatively be carried out on the filter–bank outputs.

Noise and distortions do not only effect the means of the cepstral coefficients [Openshaw and Mason 1994]. Correspondingly the second moment of the distribution, the variance can also be estimated and normalized to a fixed value (CVN). This approach is used in the RWTH feature extraction [Molau et al. 2003] and applied by many other groups e.g. [Haeb-Umbach et al. 1998, Adami et al. 2002, Pellom and Hacioglu 2003]

An overview over more sophisticated improved cepstral normalization techniques like codeword–dependent cepstral normalization (CDCN) is given in [Acero 1993]. CDCN models additive noise and convolutional distortions as as codeword-dependent cepstral biases.

Filtering Techniques

Cepstral mean normalization can be viewed as high pass filter operation with a frequency response that depends on the length of the moving window used.

Various similar filtering techniques that attempt to remove slowly varying channel biases in the logarithmic–energy of cepstrum domain have been proposed and successfully applied to increase the noise robustness. Among others there are RelAtive SpecTrAl (RASTA) filtering [Hermansky et al. 1991, Hermansky and Morgan 1994], phase–corrected RASTA [de Veth and Boves 1996] and LDA–derived RASTA filtering [van Vuuren and Hermansky 1997], high-pass filtering [Hirsch et al. 1991] and the Gaussian dynamic cepstrum representation [Aikawa et al. 1993, Singer et al. 1995]. Combinations of different methods were investigated in [Junqua et al. 1995].

Histogram Normalization

Going beyond transforming the first two moments of the distributions or linear filtering operations leads to non–linear methods that based on the complete cumulative distribution function (CDF) of the data. Histogram normalization or equalization is a standard method used in image processing [Ballard and Brown 1982] applications. The contrast of images can be enhanced by transforming the histogram of the grey scale values to make it match a given distribution. Usually a linear target CDF that corresponds to an equal distribution of the values is used.

The method was first applied in the speech processing domain to increase the robustness of a speaker recognition system [Balchandran and Mammone 1998]. The application for speech recognition tasks was suggested in [Dharanipragada and Padmanabhan 2000], were the method was used to reduce the mismatch between speakerphone and handset recordings. Later it was also successfully applied for recognition in noise e.g. [Molau et al. 2002, de la Torre et al. 2002, de Wet et al. 2003]. Section 5.2 will give a more detailed review of the histogram normalization approaches and possible improvements.

Quantile based histogram equalization [Hilger and Ney 2001] [Hilger et al. 2002] that will be discussed in this work is a parametric method that approximates the cumulative distribution function using a few quantiles, so that the method can be applied in online systems that only allow a short delay.

3.3 Adaptation Methods

Besides the feature domain approaches a corresponding adaptation of the acoustic model parameters can also be used to increase the noise robustness of automatic speech recognition system.

Training on Noisy Data

Collecting training data in the noise condition the system will be used in later, or artificially adding noise to the training data (e.g. [Gales 1995]) is a simple first step towards noise robust models. It can be effective in situations where a certain previously known background noise condition will always be present in the application. But usually the approach does not generalize in environments with different noises and as shown in the introduction to this chapter it usually affects the recognition performance on clean data.

Parallel Model Combination

Parallel or predictive model combination [Gales and Young 1996] is based on the idea of training the usual speech models and a separate noise model. In the actual recognition process the most likely combination of these two models will be calculated. This approach can effectively cope with changing noise levels that are different from those observed in training. But it is also limited in situations with various different types of background noises. Even if they are known beforehand the decision process when evaluating the most probable combination becomes more difficult. Non–stationary noises conditions also require an online update of the noise model that can be problematic if the amount of data available is not sufficient [Gales 1998].

Model Transformations

There are numerous approaches based on linear transformations that adapt given (clean) speech model parameters to the current speaker or environment conditions. An overview is presented in [Lee 1998] and in [Lee and Huo 1999].

Maximum likelihood linear regression (MLLR) [Leggetter and Woodland 1995], a matrix based transformation that was originally introduced for speaker adaptation can also be applied to increase the noise robustness [Surendran et al. 1996]. This approach was extended to maximum a posteriori linear regression (MAPLR) in [Siohan et al. 1999] by including the prior distribution of the transformation parameters and eventually structuring these prior distributions to make the method more effective if less adaptation data is available [Siohan et al. 2000].

Disadvantage of these approaches is that they require an initial recognition pass or a certain amount of adaptation from the same condition to reliably estimate the transformation matrix parameters. If online processing is required and only short estimates of the current noise condition are available the Jacobian adaptation [Sagayama et al. 1997, Sagayama 1999] approach can be used alternatively.

The influence of noise on the feature or model space can be viewed in a vector field representation. For each point in the vector field a corresponding the transformation is defined. This vector field can be described by the differential, the Jacobian matrix, that relates a change of the noise to a change of the cepstrum coefficients. Jacobian adaptation [Sagayama et al. 1997, Sagayama 1999] is based on the idea of calculating the Jacobian matrix for a certain noise condition in training. Given an estimate of current noise during recognition and its difference to the noise condition used in training the corresponding transformed mean and variance vectors can be calculated e.g. [Cerisara et al. 2000, Sagayama et al. 2001].

3.4 Likelihood Calculation

The third domain, in which a speech recognition system can be made invariant to disturbing variabilities of the signal that do not contribute to the classification, is the distance calculation domain.

Robust Decision Strategies

The usual likelihood calculation with Bayes decision rule (cf. page 5) is based on the fundamental assumption that the model parameters are a good approximation of the true distribution the data that is to be classified or recognized has. All kinds of unforeseen variabilities make this approach problematic. The decision rule can be made more robust by explicitly modelling the parameters uncertainties and modifying the classification rule [Lee and Huo 1999].

The minimax classification that was suggested in [Merhav and Lee 1993] considers the true parameter to be randomly distributed in a neighborhood region around the estimated ones. Under that assumption that this distribution is uniform the worst case probability of classification error is minimized. An application to robust digit string recognition was presented in [Jiang et al. 1998], however the extension to large vocabulary recognition has to be studied.

In Bayesian predictive classification [Huo et al. 1997] a general prior probability distribution of the parameters is assumed over which the average is calculated. The crucial problem is the estimation of the prior density and the choice of its parameters [Lee and Huo 1999].

Missing Feature Theory

The assumption that some spectral components remain reliable while others become unreliable is the basis of the missing feature theory [Cooke et al. 1996, Morris et al. 1998]. The unreliable components can be discarded [Cooke et al. 1996] or somehow replaced by corrected values [Cooke et al. 1996, Morris et al. 1998].

The problem with the approach is that it is limited to spectral features and that it depends on a reliable detection of the corrupted components [Vizinho et al. 1999]. An enhancement that are not limited to the spectral domain and does not require a detector that identifies unreliable components was suggested in [de Veth et al. 1998, de Veth et al. 2001]. A so called "local distance function" that limits the maximal distance is introduced. It makes sure unlikely feature values affect the search to a lesser degree.

Projection Based Distance Calculation

An alternative to modelling the uncertainty or neglecting unreliable feature vector components is the use of robust distance measures that are invariant to the influence of distortions on the feature space e.g. [Wood 1996]. The norm of the cepstral feature vectors

decreases in noisy conditions, as all feature vectors are pulled towards the vector of the noise [Sagayama 1999]. The orientation of the vectors is less susceptible to the distortion, based on this observation the Eucledian distance calculation can be replaced by robust projection based distances measures that take into account the angle between two vectors [Mansour and Juang 1989]. A similar approach was investigated in [Hilger and Ney 2000] were the norm of the model's mean vectors was scaled depending on the norm of the incoming feature vectors.

The tangent distance approach is more general [Simard et al. 1992]. The assumption that the norm is more affected by distortions than the angle is not made. General transformations that do not affect the class membership are considered, they define manifolds in the feature space. These manifolds can be approximated by tangent subspaces. The invariant tangent distance calculation is then based on the distance between a vector and a manifold (single sided tangent distance) or two manifolds (double sided tangent distance).

In optical character recognition applications the transformation to which the distance calculation should be invariant are known [Keysers et al. 2000a, Keysers et al. 2000b]: e.g. line width, scaling and rotation. This kind of prior knowledge of the transformations can be used to determine the tangents, but it is not required. The invariances can also be estimated from the training data, allowing the application in speech recognition systems were comparable prior knowledge of the invariances is not obvious [Macherey et al. 2001].

The adaptation methods (section 3.3) and the modified likelihood calculation (section 3.4) approaches require modifications of the recognition system that shall not be investigated in this work. Here the focus will be put on the optimization feature domain approach that can be added to an existing system, without requiring specific modifications of the recognizer itself.

Chapter 4

Scientific Goals

Within this work a method to increase the noise robustness of automatic speech recognition systems shall be introduced and explored in detail. Some of the limitations, restrictions and disadvantages the afore mentioned existing methods have, shall be overcome with this method. The requirements for the method are based on conclusion drawn from considerations about other approaches that can be summarized as follows:

- **Feature domain:** the method shall be applied in the feature extraction front–end. It shall not require any feedback from – or interaction with the recognizer used as back–end, so it can be added to an existing system without requiring modifications of the recognition engine. In principle, there should even be the possibility to add the method to the front–end and thus increase the noise robustness without making a retraining the system necessary. In a distributed speech recognition scenario with the feature extraction on mobile terminals (e.g. telephones or PDAs) and server side speech recognition, like it is studied within the Aurora project [ETSI 2000, Pearce 2000], such an approach is advantageous. The noise robustness of individual new terminals could be enhanced by adding the new method, without requiring an update of the complete system, i.e. all terminals and the servers.

- **Independence from the recognition task:** the approach should not require mayor parameter optimizations when applying it to a new data set and it should be independent from the complexity of the task, which means that it should work together with low complexity small vocabulary speech recognition systems and more complex large vocabulary recognition applications alike.

- **Single channel:** stereo data, microphone arrays, audio–visual recording or other special hardware setups should not be required for the method. It should already work on single channel recordings with an arbitrary sampling rate. The comparably good human recognition performance on noisy single channel recordings show that there still is room for large improvements of automatic systems that can be explored before resorting to modified recording hardware setups.

- **Independence from the type of training data:** the approach should not rely on the availability of clean — or specific kinds of task dependent noisy training

data, respectively isolated recordings of sample noises. Some of the afore mentioned methods require clean training data to estimate the clean speech distribution that will be used as target for a transformation. This is a disadvantage since noisy training data usually provides a better recognition performance, if the system is to be primarily used for the recognition of noisy data. On the other hand no prior information or assumptions about typical noises or SNRs to be expected during recognition should be required either. Even if no data to train a typical noise model for the final application is available and the SNR of the typical test data is not known during the training of the system, the approach should still work well.

- **Without speech silence detection:** the feature extraction should not forestall the decision process of the recognizer. A hard, irreversible decision about removing non speech samples from the data should not be taken by the front end. And even if the non speech frames are not removed, relying on a speech silence detection for the noise estimation during the feature extraction can be problematic, especially when the SNRs are low. So the approach should consider the data's distribution as a whole without out distinguishing between speech and silence frames or carrying out any other classification.

- **Single pass online method:** practical speech recognition applications that have to cope with adverse environment conditions like name dialing, command and control or dialog systems require a system response in real–time. An approach that can be used for this kind of application should be a single pass method that is suited for a moving window implementation with a small delay. It should also be able to work without additional adaptation data, because in this kind of application the noise environment can not be expected to be stationary over more than one or a few utterances. So, even if there is only one utterance that consists of one single word, the approach should be able to work.

- **Computationally inexpensive:** besides the delay the algorithm has in principle the system response is determined by the computational complexity, so the approach should be computationally inexpensive and work with little memory requirements.

The following chapters will show how quantile based histogram equalization can meet these demands. It is an approach to increase the noise robustness by reducing the mismatch between the training and test data distributions with a parametric transformation function. It goes beyond simple mean and variance normalization without, requiring the long delay non–parametric transformations based on full histograms need.

Chapter 5

Quantile Based Histogram Equalization

5.1 Introduction

Before going into the actual details of quantile based histogram equalization an example shall illustrate how the general concept of mismatch reduction motivates transformations based on the cumulative distribution functions. A short review of non–parametric transformations with full histograms will then be the starting point for the introduction of quantile based histogram equalization.

The mismatch between clean and noisy signals can be viewed and eventually reduced at different positions in the feature extraction. In the example that shall be discussed here, the output of the Mel–scaled filter–bank shall be considered after applying a non–linear function to reduce the dynamic range (figure 5.1). Usually a logarithm is applied at this point, but as it will be shown later in detail, replacing the logarithm by a root function can make the recognizer more noise robust. So figure 5.2 on page 36 shows the output of the Mel–scaled filter–bank after applying a 10th root compression.

The example is taken from the Aurora 4 noisy Wall Street Journal database, it shows the initial 5 seconds from the last sentence in the test set. The utterance is: "The index ended with a decline of zero point three five ... " (447c0216.wv1). The output of the 6th filter (of 20 in total) for the original clean signal (taken from test set 1) is plotted together with the signal distorted by street noise and a microphone mismatch (test set 12). Especially in the speech pauses the mismatch between the two signals is large, due to the background noise. There also is a difference between the speech portions of the signal, but it is much smaller. The high speech peaks of the signal stick out and are not covered by the background noise.

This example can be used as a motivation for different shifting or scaling methods that reduce the mismatch of the two signals. Going back to the spectral spectral or Mel–filter domain before the logarithm respectively root, a spectral subtraction would correspond to a shifting of the signal, while the application of a Wiener filter would be a rescaling. These methods depend on a reliable speech silence estimate to determine the noise estimate that is subtracted or used when calculating the gain. Other shifting and scaling methods do

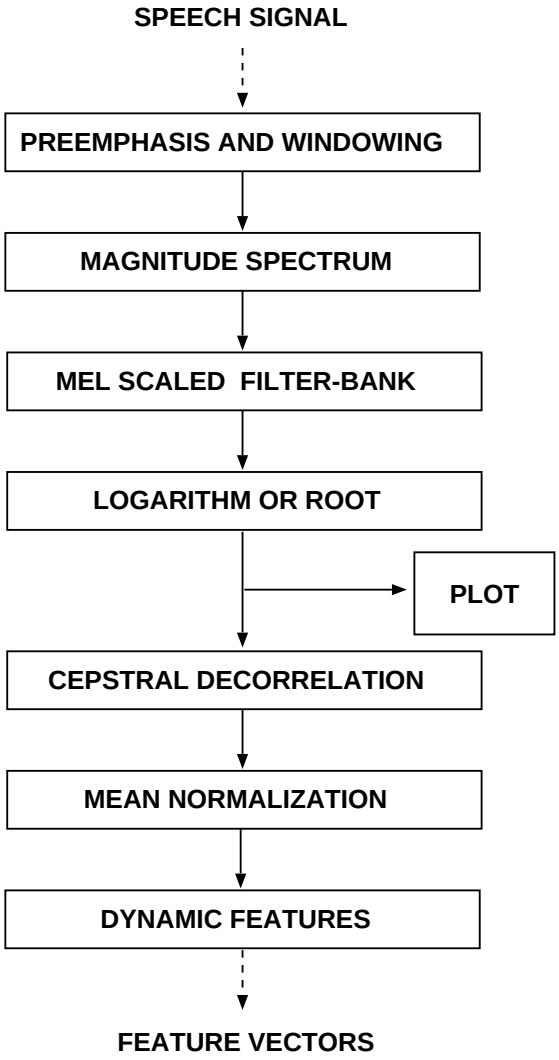

Figure 5.1: Feature extraction front end showing the position of the output used to plot the signals in figure 5.2.

not explicitly distinguish between speech and silence, the overall statistics of the signal can be considered instead: e.g. mean and variance normalization in the logarithmic or root compressed domain are simple but efficient linear transformation methods to reduce the mismatch. They are used as standard in many recognition systems.

The mean and the variance are the first two moments of a distribution, so an obvious generalization would be to consider the data's distribution as a whole and base the reduction of an eventual mismatch on this distribution. Figure 5.3 shows the cumulative distribution functions corresponding to the signal plotted in figure 5.2.

The slope of the clean distribution is typical. It is steep in the low amplitude region, there are many small values within a small range during the silence portions of the signal. Then, in the higher amplitude speech portions the slope is less steep. The observed speech frame values lie within a larger range. The corresponding distribution of the noisy signal

again clearly shows how the noise affects the low amplitude speech pause regions of the signal more than the actual speech regions.

Two cumulative distribution functions like the ones shown in figure 5.3 can be used to define a transformation that maps the distribution of a noisy input signal back to the distribution of a clean signal. In a real application the clean reference signal and the corresponding cumulative distribution for the specific spoken sentence is obviously not available, so it has to be replaced by the overall cumulative distribution of the training data. This idea is the basis of the full histogram normalization.

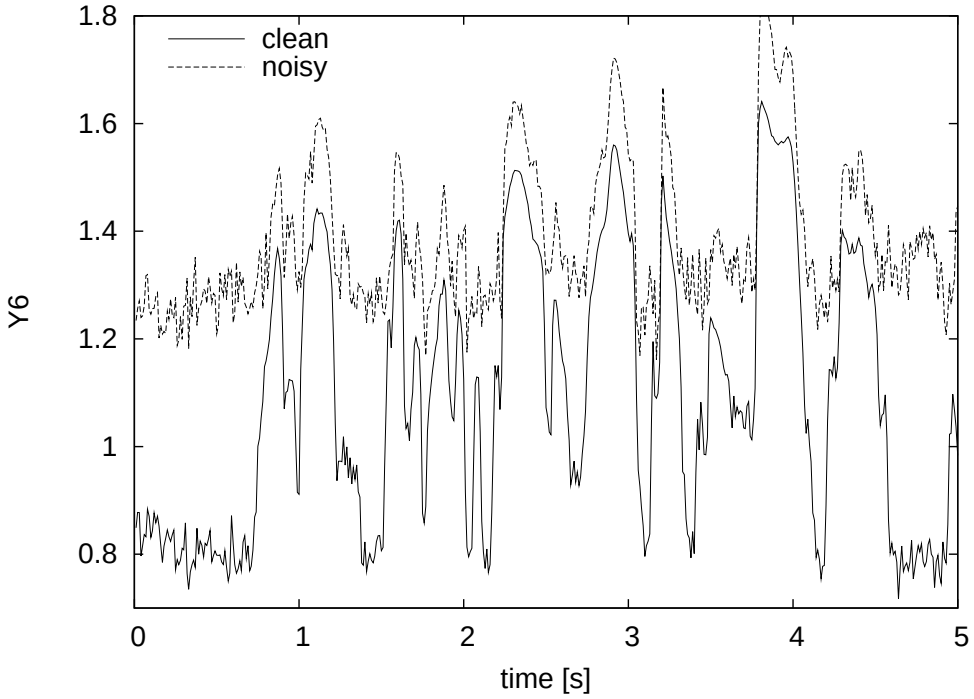

Figure 5.2: Example: output of the 6th Mel scaled filter over time for the last sentence from the Aurora 4 test set (447c0216.wv1).

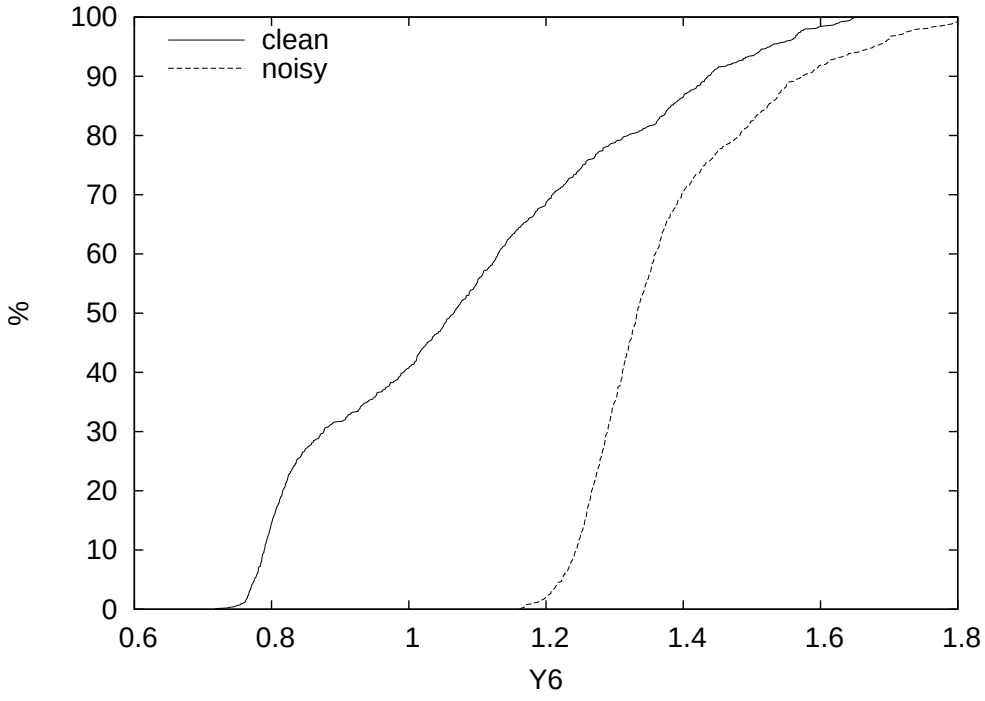

Figure 5.3: Cumulative distributions of the signals shown in Figure 5.2.

5.2 Histogram Normalization

Histogram normalization is a general non–parametric method to make the cumulative distribution function (CDF) of some given data match a reference distribution. It is a common method in image processing e.g. [Ballard and Brown 1982] were it is usually used to enhance the dynamic range and contrast of the images, but it is not limited to this kind of image processing application. As the example in the previous section has illustrated, it can also be used in speech processing to reduce an eventual mismatch between the distribution of the incoming test data and the training data's distribution which is used as reference.

In [Balchandran and Mammone 1998] it was applied in a speaker identification task. A CDF based transformation was applied to the samples, to increase the robustness of the identification on distorted utterances. The application for speech recognition was suggested in [Dharanipragada and Padmanabhan 2000]. The feature vector components were transformed to reduce the mismatch between speakerphone and handset recordings. Other publications have confirmed the usefulness of the approach especially in noisy conditions.

The transformation can be applied at different stages of the feature extraction. The use in the Mel scaled filter–bank domain was suggested in [Molau et al. 2001]. The application to cepstral coefficients was described in several publications [de la Torre et al. 2002, Segura et al. 2003, de Wet et al. 2003] and the approach was extended to the transformation of the cepstral derivatives in [Obuchi and Stern 2003].

In the following the general concept behind non–parametric transformations based on full histograms shall be considered: provided that enough data (some minutes) from the current acoustic condition is available, detailed cumulative histograms can be estimated without leaving empty histogram bins. If the spoken phonemes and words in the utterances used to estimate the histograms are diverse enough, the assumption holds that the global statistics of the speech data are representative. In that case the only remaining systematic mismatch between the test and the training data distributions is caused by the different acoustic conditions and not by what was actually spoken [Molau et al. 2002], thus the two CDFs can be used directly to define a transformation.

If P is the CDF of the current test data and $P^{train^{-1}}$ the inverse reference CDF of the training data the transformation of an incoming value Y simply is and [Dharanipragada and Padmanabhan 2000, Molau et al. 2001, de Wet et al. 2003] :

$$\hat{Y} = P^{train^{-1}}\left(P(Y)\right) \tag{5.1}$$

Instead of using the training data distribution as target a Gaussian with zero mean and unity variance can also be used as target probability distribution [de la Torre et al. 2002].

In both cases these expression can be implemented as simple, computationally inexpensive table look–up, if the resolution of the histograms is large enough. For each incoming value the closest value in the table is chosen and the corresponding output is passed on to the next step of the feature extraction. This transformation is non–parametric, there is no parameter that actually describes the shape of the transformation function. The only parameter that has to be defined is the resolution of the histogram.

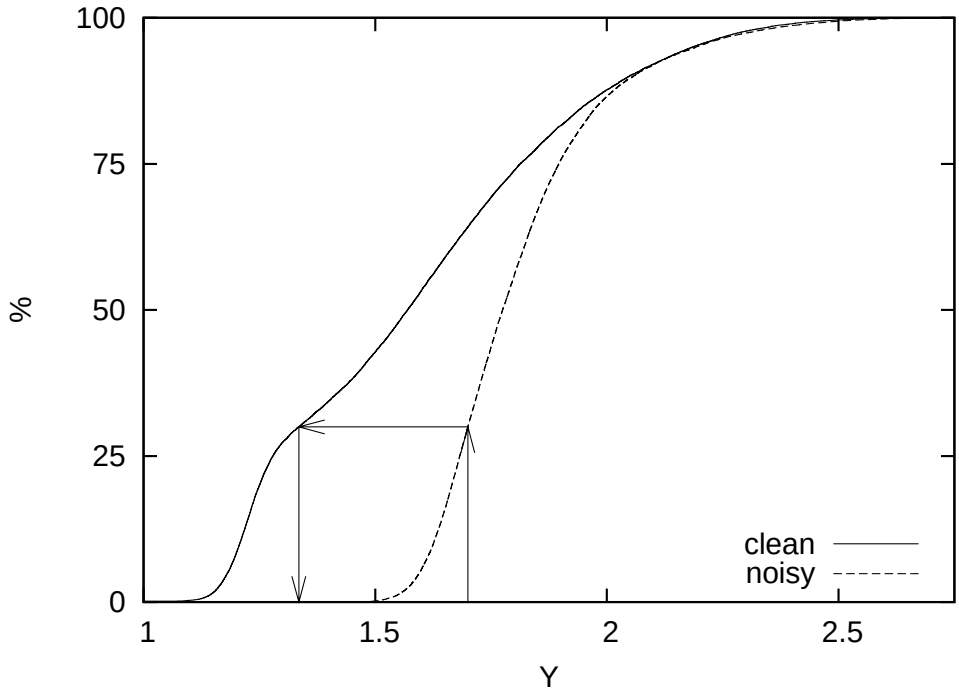

Figure 5.4: Example for the cumulative distribution functions of a clean and noisy signal. The arrows show how an incoming noisy value is transformed based on these two cumulative distribution functions.

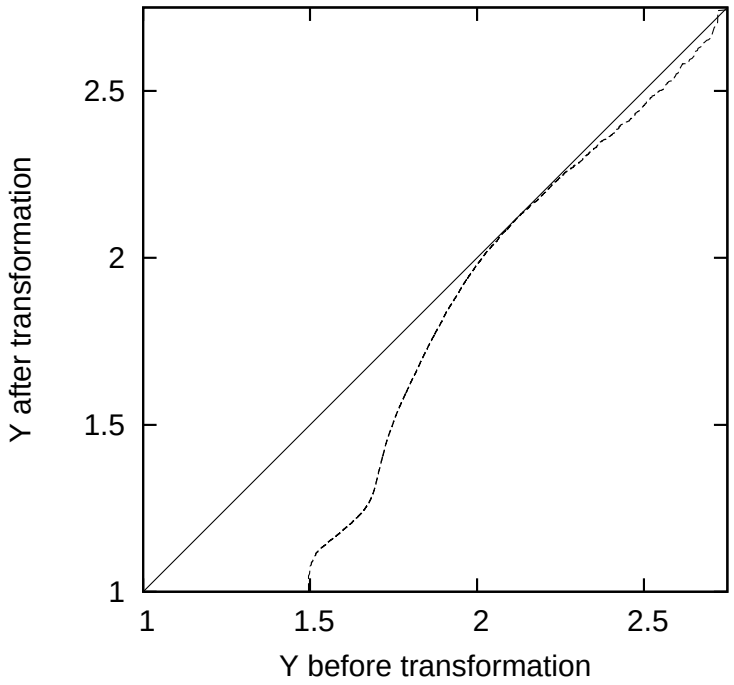

Figure 5.5: Transformation function based on the full cumulative distributions shown in figure 5.4.

The process of calculating a new output value for a certain incoming value based on two cumulative distributions is illustrated in figure 5.4. The actual transformation function i.e. equation 5.1 that is the result of this mapping procedure is depicted in figure 5.5.

The derivative of the cumulative distributions, i.e. the probability distribution functions of the speech data are usually bimodal, the contribution of silence and speech to the distribution can be distinguished. This is the basis for two very effective optimizations of the algorithm. The transformation function can be smoothed by fitting a bimodal Gaussian function to the probability histogram [Molau et al. 2001] and the target distribution can be adapted to the amount of silence in the current utterance [Molau et al. 2002]. Two separate histograms, one for silence the other for speech, can be estimated on the training data. Then a first recognition pass can be used to determine the amount of silence in the recognition utterances. Based on that percentage the appropriate target histogram can be determined. A detailed description of these methods is given in [Molau 2003].

With regard to many applications the disadvantage of the histogram normalization method is that it is a two pass method, which requires a sufficiently large amount of data from the same recording environment or noise condition to get reliable estimates for the high resolution histograms. It can not be used when a real–time response of the recognizer is required, like in command and control applications or spoken dialog systems. And even if real–time response is not the crucial issue, the application of full histogram normalization can be problematic if the acoustic conditions change significantly from one utterance to the next.

In these cases a method is required that provides a sufficiently reliable estimation of the transformation function, even with little data. A straight forward solution to this problem would be to reduce the number of histogram bins, in order to get reliable estimates even with little data. Then a linear interpolation between these histogram bins could be applied. But this approach has some disadvantages too: the range and the distribution of the data has to be determined to define an adequate spacing of the bins and the optimal spacing can change with the acoustic conditions recording conditions.

Giving up the non–parametric transformation and the fixed histogram bins shall be considered in the next section when quantile based histogram equalization is introduced. Which is an approach that uses a parametric transformation function with few parameters that can be reliably estimated independent from the amount, scaling and range of the incoming recognition data.

5.3 Quantile Based Histogram Equalization

Many practical speech recognition applications require a system response in real–time and the capability of the system to cope with quickly changing acoustic environment conditions. If the noise reduction in such a system shall still be based on reducing the mismatch between the current data distribution and the training reference, it must be able to get a good approximation of these distributions from some seconds of data or in some cases even isolated words.

A commonly used approach that can be used if only small amounts of data are available is the estimation of the mean and the variance, the first two moments of a distribution, to carry out a mean and variance normalization. A possible generalization of this approach would be to also include the third moment of the distribution, the skewness, and eventually the higher moments.

Here quantile based histogram equalization ("quantile equalization" QE) shall be considered as alternative. Cumulative distributions can be approximated using quantiles [Gnanadesikan 1980]. Quantiles are very easy to determine by just sorting the sample data set. Given a factor $q \in [0, 1]$ the corresponding quantile is defined as the value in the set that is larger than the fraction q of the values. It can be determined by simply picking the entry at position $[q \cdot N]$ in a sorted list with N entries. The 50% quantile is also known as the median of the distribution. Figure 5.6 shows an example, two cumulative distribution function with four 25% quantiles, $N_Q = 4$ [Hilger and Ney 2001, Hilger et al. 2002]. Like in the introduction to this chapter the data for the example was taken from the Mel–scaled filter–bank (figure 5.1 on page 34), but all considerations described here are general and not restricted to the filter–bank domain.

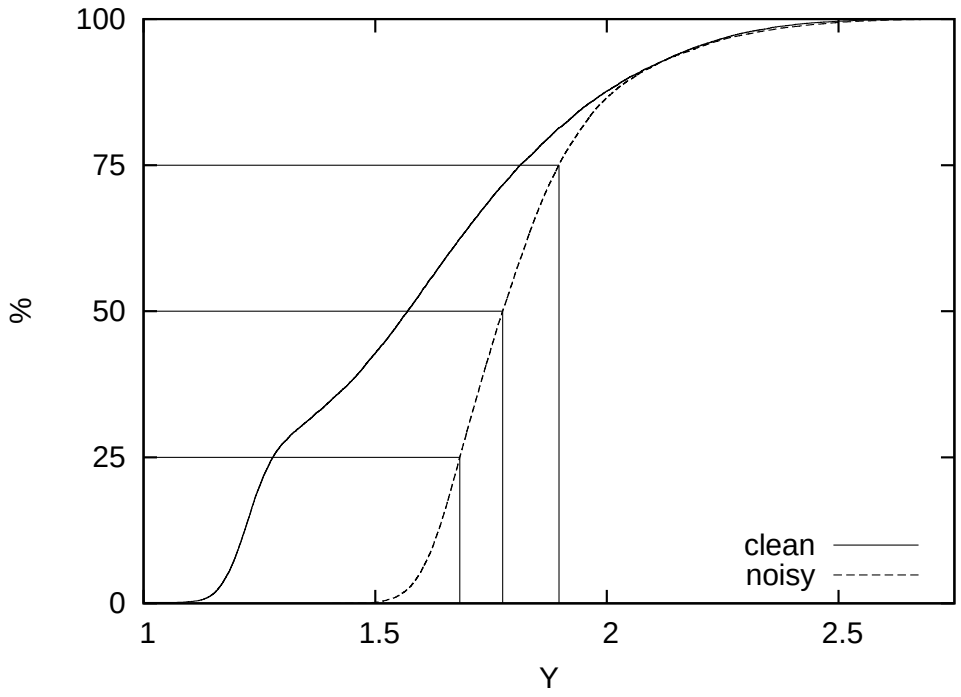

Figure 5.6: Two cumulative distribution functions with 25% quantiles.

By the way they are defined, the quantiles provide a robust approximation of the cumulative distribution function that is completely independent from the scaling and range of the data. A histogram with fixed bins on the amplitude axis would have to be readjusted if the data's scaling changes significantly, the process of picking entries from a sorted list instead is not affected by the scaling. The quantiles automatically adapt to changes of the scaling.

The independence from the amount of data is an other advantage of the quantile approach. Even if the data set that shall be considered only consists of very few or in an extreme case just one sample, the quantiles can be calculated without any special modification of the algorithm. Of course in that case the quantiles will not provide a reliable estimation of the distribution yet — but as long as the total number number of quantiles that shall be calculated is kept small e.g. $N_Q = 4$, like shown in the example, about one second of data (100 time frames) is already sufficient to get a rough estimate of the cumulative distribution that can serve as basis for the transformation of the data.

The reference quantiles that will later serve as target for the transformation are calculated on the training data of the system. Whether it is clean or noisy does not affect the algorithm described in the following. Q_{ki}^{train} shall be the ith training quantile ($i \in [0, \ldots, N_Q]$) for the kth vector component. The correct method of storing the entire training corpus to sort it and determine the quantiles is not practicable. The approach used instead is determining the quantiles utterance by utterance and then calculating average quantiles from these values.

Taking the averaging one step further the training quantiles can be pooled over the filter channels to get Q_i^{train} independent from the component index k. Even though the typical overall amplitude distributions do differ in the individual filter channels, leading to significantly different filter specific training quantiles, the pooling does usually not affect the recognition performance (section 6.4.1 on page 101 and [Hilger and Ney 2001, Hilger et al. 2002]). After their estimation these reference quantiles could be used to transform the training data in a second pass before actually training the system, but experiments (section 6.4.3 and [Hilger et al. 2003]) show that this step is not necessary. Transforming the training data usually does not provide better recognition results in the end.

During recognition process the quantiles Q_{ki} are determined on the current utterance. To avoid scaling up noises which are lower than the average level observed in training, lower bounds for the recognition quantiles Q_{ki} can be defined:

$$\text{if} \quad Q_{ki} < Q_i^{train} \quad \text{then} \quad Q_{ki} = Q_i^{train} \tag{5.2}$$

These recognition quantiles do have to be vector component, i.e. filter channel k specific to be able to cope with the different spectral characteristics of occurring noises. If they were pooled too, the method would only be able to cope with white noise.

The recognition quantiles combined with the corresponding reference quantiles of the training data define a set of points that can be used to determine the parameters θ of a transformation function T that transforms the incoming data Y to $\tilde{Y}$ and thus reduces the mismatch between the test and training data quantiles (figure 5.7):

$$\tilde{Y} = T(Y, \theta) \tag{5.3}$$

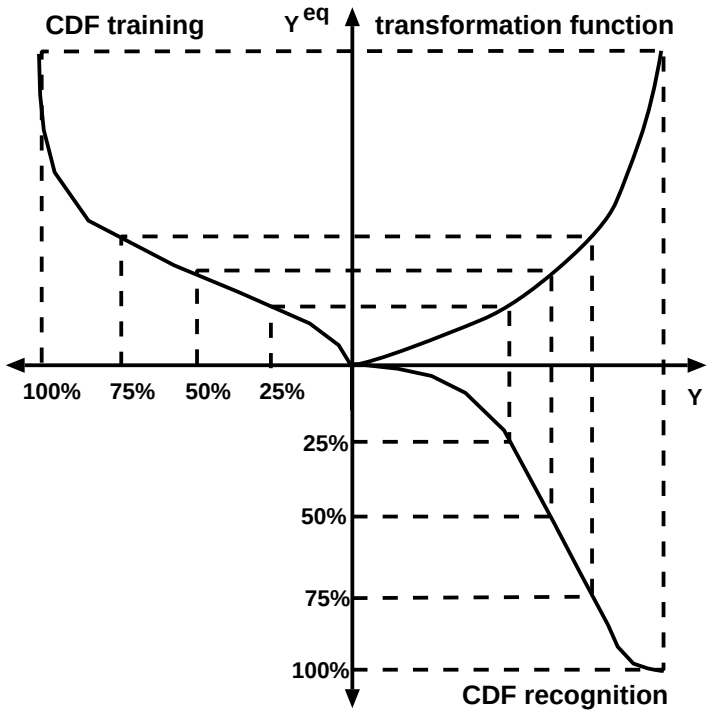

Figure 5.7: Applying a transformation function to make the four training and recognition quantiles match.

The concept of using a parametric transformation function at some appropriate stage of the feature extraction is very general. Any function that reduces the mismatch between training and test data without distorting the signal too much could be used and might improve the recognition performance.

The question arises where in the feature extraction the transformation should be applied and what its actual functional form should be? In principle it could already be applied on the power spectrum. But calculating the quantiles and transformation parameters for some hundred spectrum coefficients is a waste of computational resources, since the dimensionality of the features will be significantly reduced in the following feature extraction steps.

Within the context of this work the transformation is applied to the output of the Mel–scaled filter–bank after applying a 10th root to reduce the dynamic range, so in the following Y will denote the output vector of the filter–bank and Y_k will correspondingly denote its kth component. Note that Y_k is a function of time t but an additional index for t will not be introduced to keep the equations more readable.

The use a root function instead of the logarithm was originally only based on the idea of having zero as fixed lower limit for the values after the reduction of the dynamic range. A lower limit that is not transformed can be used as fixed point for the transformation function. As positive side effect observed in the experimental evaluations, it turned out that the use of a root function can already improve the recognition on noisy data. The general relation between root/power functions and the logarithm can be expressed as

follows:

$$f_r(x) = \frac{x^r - 1}{r} \tag{5.4}$$

A comparison between the Taylor series expansion for this expression and the one for the logarithm reveals that the limit of $f_r(x)$ for $r \to 0$ is the logarithm:

$$\lim_{r\to 0} f_r(x) = \lim_{r\to 0} \left((x-1) + \frac{1}{2}(r-1)(x-1)^2 + \dots \right) = \log(x) \tag{5.5}$$

A special property of the logarithm is that a constant gain applied to the input will result in a simple shift of the output. Such a shift, typically introduced by the transmission channel, can be eliminated by mean normalization i.e. a subtraction of the longterm mean. This nice property is lost when using $r > 0$, but there is experimental evidence that this is no drawback. On the contrary, the noise robustness can be increased when replacing the logarithm by an appropriate root function. The experiments presented in [Tian and Viikki 1999] show that a value of r around 0.1 gave best recognition results in noisy conditions.

The approach of replacing the logarithm by a root can be generalized even more: the constant shift of -1 and scaling by $1/r$ in equation 5.4 will both be applied during training and recognition, so they will not affect the final recognition result. Thus an expression of the type x^r can be used instead for the actual application [Hilger et al. 2003]. The detailed experimental investigation on the effect of different values of r on clean and noisy data presented in section 6.2.2 show confirm that root functions can even outperform the logarithm on noisy data.

After defining the domain in which the transformation shall be applied, the functional form of the transformation function T has to be defined. The straight forward approach would be to use a piecewise linear transformation function that simply connects the points defined by the test and training quantiles.

This very simple method has a disadvantage that was confirmed in recognition experiments [Hilger and Ney 2001]: the slope of the transformation function can change significantly from one piecewise linear segment to the next, which will lead to distortions of the output signal.

When using an appropriate non–linear function with continuous derivative this problem can be avoided. The properties of such a transformation function T (equation 5.3) have to be tailored to meet the demands defined by the signal's characteristics that were illustrated in the example (figure 5.2 on page 36):

- The signals are positive, the origin is fixed and shall not be transformed.
- Small amplitude values suffer from significant distortions by noise, they have to be scaled down considerably, to bring them back to the level of a clean signal.
- The mismatch decreases towards higher amplitudes, the highest values require just a little or no scaling.

For values within the interval [0,1] a power function of the form x^γ with $\gamma > 1$ has the required properties. The origin is fixed, the value 1 is not transformed either, and in

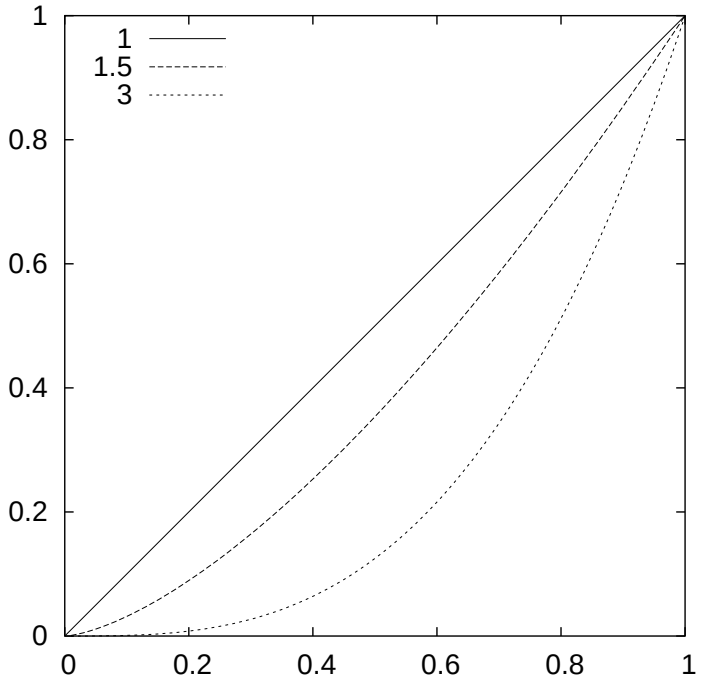

Figure 5.8: Example of a power function transformation x^γ for different values of gamma.

between small values are scaled down more than the higher ones. To make clear that this exponent is different from the $r < 1$ that was used previously for the root compression it is denoted γ, which is an allusion to gamma correction that is used to adjust the brightness in image processing applications [Poynton 1993].

To scale the incoming filter output values Y_k down to the interval $[0, 1]$ they have to be divided by the maximal value, which simply is the largest quantile Q_{kN_Q}. On some databases the recognition performance can be improved by allowing a certain transformation of the highest value, so in this case an an overestimation factor o between 1.0 and 1.5 can be introduced. Then $o \cdot Q_{kN_Q}$ should be used instead of just Q_{kN_Q}. After the power function transformation is applied the values are scaled back to the original range:

$$\tilde{Y}_k = T_k(Y_k, \theta_k) = Q_{kN_Q} \left(\frac{Y_k}{Q_{kN_Q}} \right)^{\gamma_k} \tag{5.6}$$

This type of function is sufficient for the transformation of filter outputs after root compression, but if the function shall be generally applicable at different stages of the feature extraction, i.e. also in the spectral domain as well as the filter–bank domain before (logarithmic) compression (section 6.4.4 and [Hilger and Ney 2001]) the problem arises that its derivative is zero at the origin. Small values are scaled down even further towards zero, so little amplitude differences will be enhanced considerably if a logarithm is applied afterwards, this is in contradiction to the desired compression of the signal to a smaller range. This unwanted effect can be reduced by introducing an additional term that dominates the expression near zero and reduces the relative transformation. A linear term with a constant constant derivative can take this role, so the transformation function that will always be used within the context of this work (unless stated otherwise) is:

$$\tilde{Y}_k = T_k(Y_k, \theta_k) = Q_{kN_Q} \left(\alpha_k \left(\frac{Y_k}{Q_{kN_Q}} \right)^{\gamma_k} + (1 - \alpha_k) \frac{Y_k}{Q_{kN_Q}} \right) \tag{5.7}$$

The second parameter α_k is a weight factor that balances the influence of the power function and the identical transformation. Both transformation parameters $\theta_k = \{\alpha_k, \gamma_k\}$ are jointly optimized to minimize the squared distance between the current quantiles Q_{ki} and the training quantiles Q_i^{train}:

$$\theta_k = \underset{\theta'_k}{\operatorname{argmin}} \left(\sum_{i=1}^{N_Q-1} \left(T_k(Q_{ki}, \theta'_k) - Q_i^{train} \right)^2 \right) \tag{5.8}$$

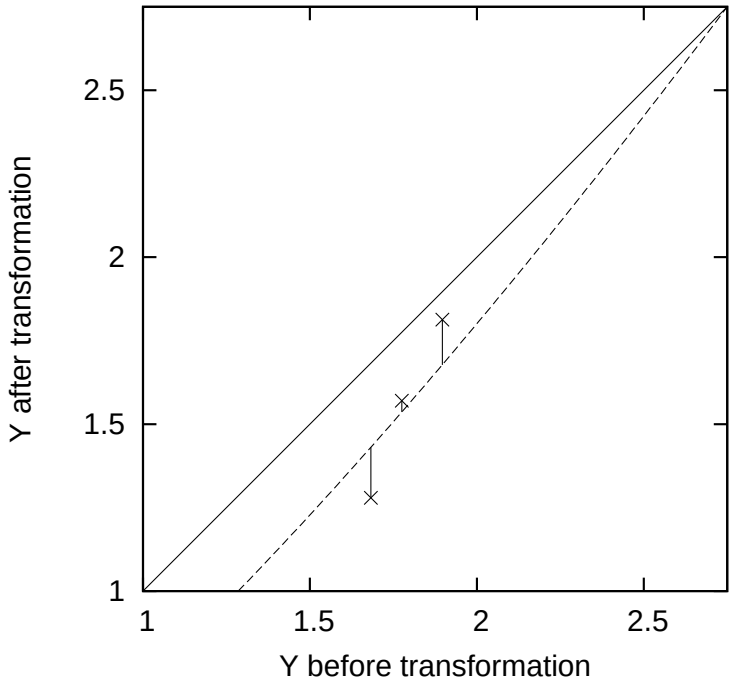

Figure 5.9: Parametric transformation based on the quantiles shown in Figure 5.6. The points (Q_i, Q_i^{train}) are marked by $\times$. The parameters of the transformation function are chosen to minimize the squared distance between the clean and the transformed noisy quantiles.

In equation 5.8 the lowest and highest quantiles ($i = 0$ and $i = N_Q$) are not considered in the sum because these outlier values should not affect the estimation of the transformation parameters. The minimum is determined with a simple grid search: α_k, by the way it is defined in equation 5.7, should be in the range $\alpha_k \in [0, 1]$. The exponent γ_k should be larger than 1 to actually scale down the incoming values and by limiting it to a maximal value of e.g. $max = 3$ the maximal distortion of the signal can restricted which generally leads to better recognition results, so $\gamma_k \in [1, max]$. The step size for the grid search can be set to a value in the order of 0.01. Usually smaller grid sizes only effect the

computational requirements, without improving the recognition results. Which indicates that other more sophisticated optimization methods are also not likely to find a better optimum that makes a difference in terms of recognition results.

Figure 5.9 illustrates the minimization process based on the example quantiles shown in figure 5.6 page 40. The solid diagonal line represents identical transformation: each incoming value "before transformation" and outgoing value "after transformation" are the same. The points marked with $\times$ represent the quantiles, their coordinates are $\{Q_i, Q_i^{train}\}$. The distance between the training and recognition quantiles corresponds to the vertical distance between the point $\times$ and the transformation function that is applied. The dashed line is a transformation function (equation 5.7) after optimization. Its overall distance to the points is smaller than the distance of the simple identical transformation.

The power function transformation will minimize the difference between training and test quantiles. The shape of the distribution is changed, but it can not be guaranteed that the test and training quantiles are exactly identical after the transformation, so quantile equalization will not be able to replace the standard mean normalization. With regard to the online implementation that will be introduced in the next section the mean normalization will be carried out directly after quantile equalization in the filter–bank domain (figure 5.10) and not as usual in the cepstral domain. Since the cepstrum transformation is a linear transformation this repositioning should not have any mayor effect on the recognition.

The figures 5.11 and 5.12 illustrate the influence of quantile equalization on the example that was already used in the introduction to this chapter on page 36. The transformation parameters (equations 5.6) that were found to be optimal by the grid search were $\gamma = 1.4$ and $\alpha = 1.0$ in this case. Plotting the signal over time (figure 5.11) shows how the background noise level is clearly pulled down towards the original level of the clean signal, while on the other hand the speech portions of the signal are not transformed as much.

The transformed cumulative distribution still does not match the original clean one perfectly (figure 5.12), this could not be expected from the simple power function transformation that was applied, but the mismatch is obviously reduced. Especially in the region above amplitudes of 1 the resulting CDF is close to the original clean reference.

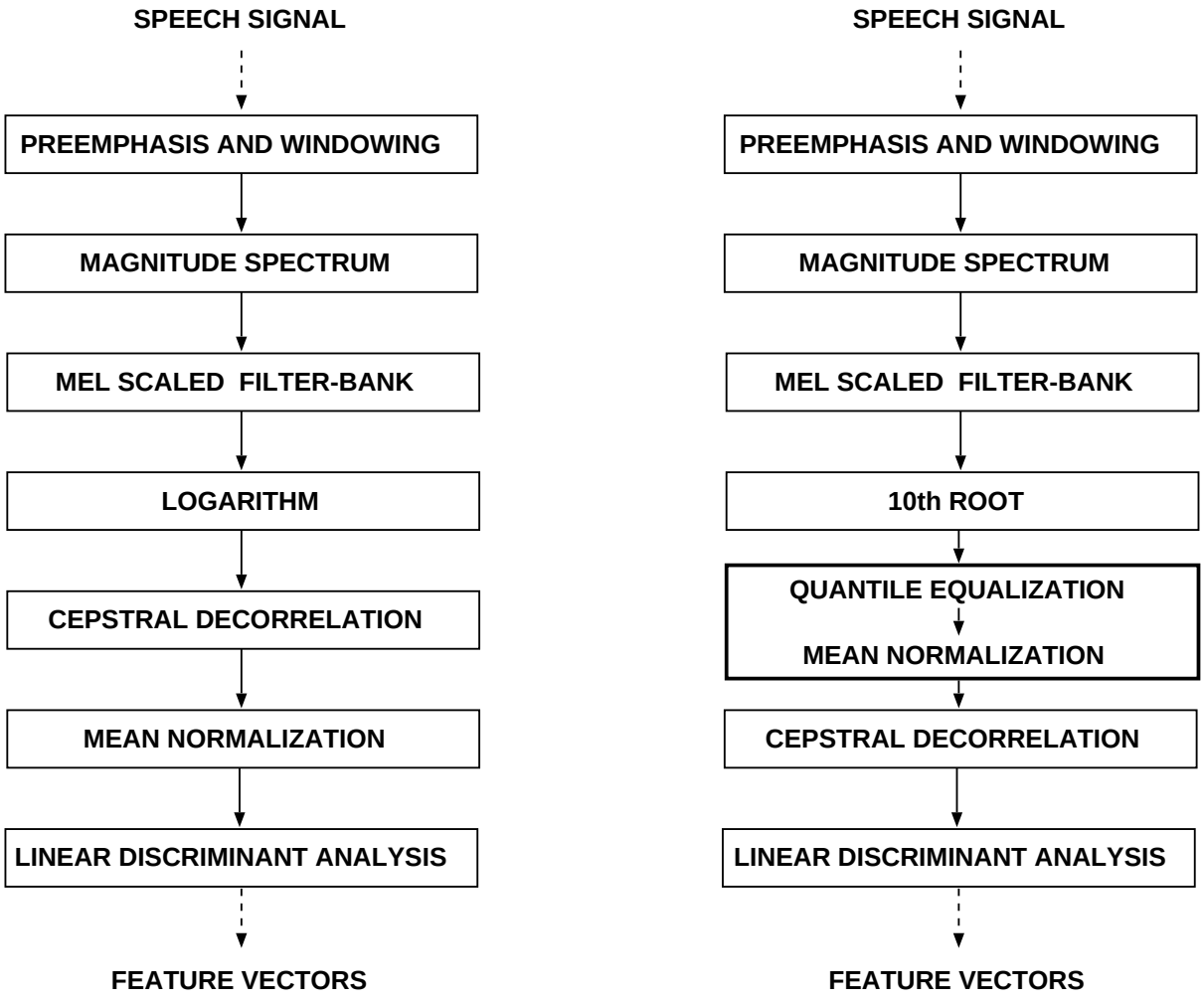

Figure 5.10: Comparison of the RWTH baseline feature extraction front–end and the version with 10th root compression, quantile equalization and joint mean normalization.

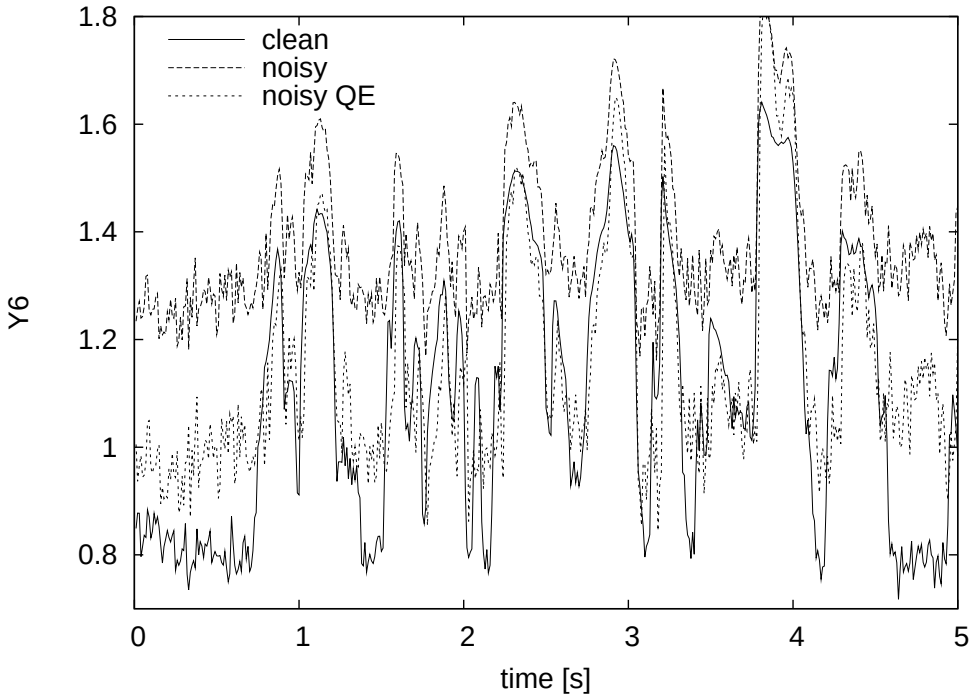

Figure 5.11: Example: output of the 6th Mel scaled filter over time for a sentence from the Aurora 4 test set before and after applying utterance wise quantile equalization.

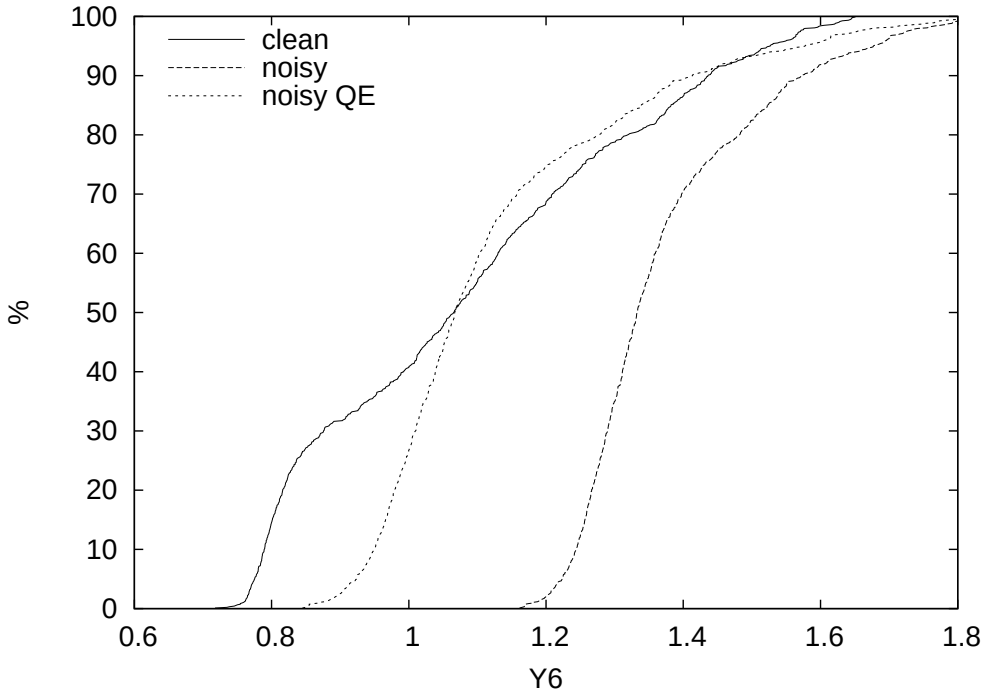

Figure 5.12: Cumulative distributions of the signals shown in Figure 5.11.

5.4 Online Implementation

In the previous section the assumption was that the quantiles are determined on an entire utterance and the transformation parameters are calculated once to remain constant for that utterance. This restriction can be dropped, but the real online application with a moving window requires some more considerations [Hilger et al. 2002] that will be discussed in the following.

For online applications it is standard to implement mean and variance normalization using a moving window. If the delay and window length are chosen appropriately the recognition performance will not suffer significantly.

Quantile equalization can also be implemented using a window instead of the whole utterance, but when when simply applying the two techniques successively their individual delays add up as shown in Figure 5.13. An initial delay has to elapse before the quantile equalization passes the first vector to the mean normalization, then the second delay of the mean normalization has to go by before the first vector is actually put out and the feature extraction can continue with the calculation of the cepstrum coefficients.

Figure 5.14 illustrates an alternative that combines the two steps without adding the delays [Hilger et al. 2002]. Assuming that quantile equalization and mean normalization have the same delay, the resulting delay is halved with this procedure, at the cost of a growing the computational complexity.

For each time frame t:

1. Calculate the signal's quantiles Q_{ki} for each filter channel in a window around the current time frame. The window length t_{win} should be some seconds. It does not have to be symmetrical. The delay t_{del} can be chosen to be short (some time frames) if the application only allows short delays or longer (seconds) if the recognition performance is more important.
2. If $Q_{ki} < Q_i^{train}$ then $Q_{ki} = Q_i^{train}$
3. Determine the optimal transformation parameters α_k and γ_k and apply the transformation to all vectors in the window.
4. Calculate the mean values of the resulting vectors within the window.
5. Subtract the mean to get the final vector of filter bank coefficients.

After that step the feature extraction can be continued as usual with the calculation of the cepstral coefficients.

In the online implementation the expression in step 2. does not only make sure that a noise level with lower amplitude than in training is not scaled up, it also provides an important initialization of the quantiles at the beginning of the utterance: if the moving window only contains non–speech frames at the beginning of the utterance even the high quantiles will be determined by these silence frames. A transformation that is simply be based on this estimate would then transform the background noise level to the speech

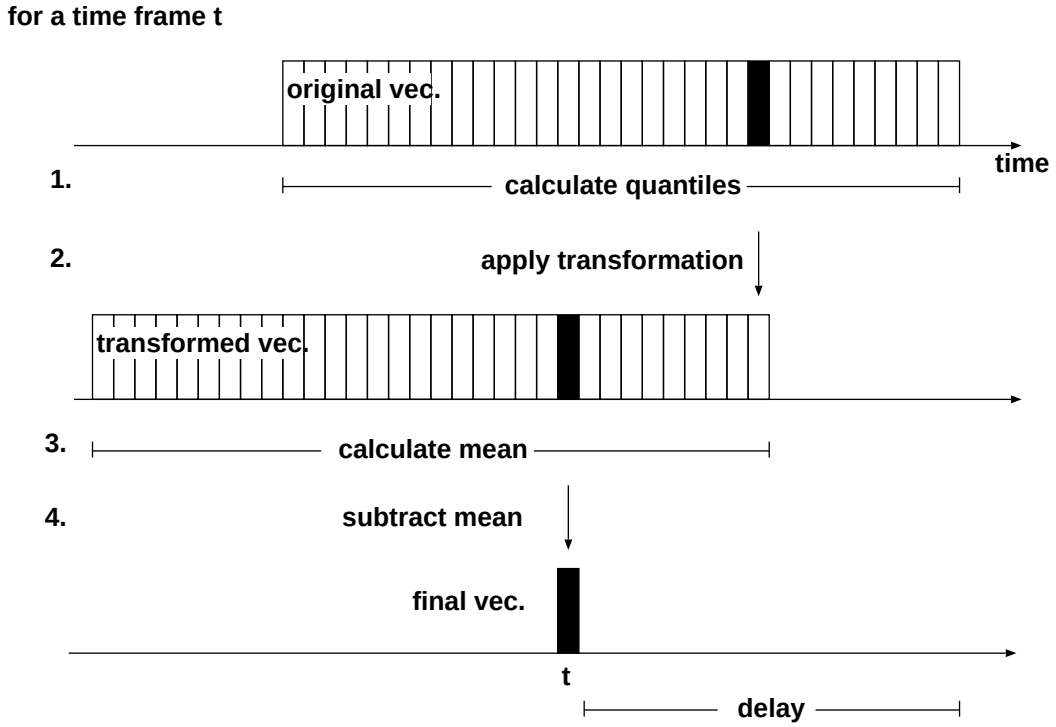

Figure 5.13: Application of quantile equalization and mean normalization using two successive moving windows, both delays add up.

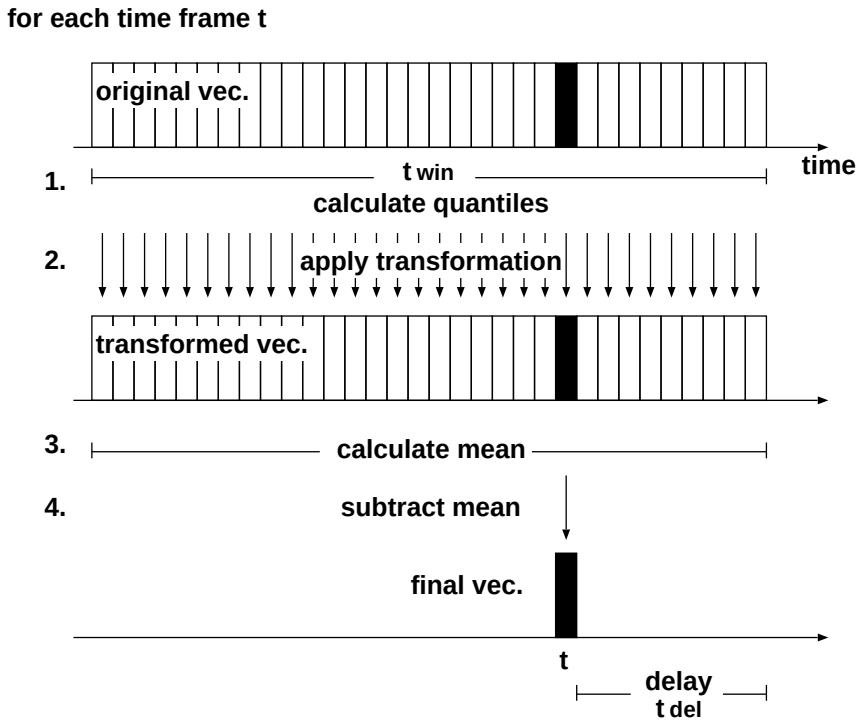

Figure 5.14: Combined normalizing scheme with shorter delay.

level observed in training. By initializing the quantiles according to expression 2. this can be prevented. As long as the SNR is not too low the background noise level will not exceed the amplitude of the speech peaks, so the higher training quantiles can take the role of the speech estimate if that is not available yet.

The update of the parameters α_k and γ_k in step 3. requires some modifications to make it practically applicable in an online system. When using a full grid search as described in Section 5.3 in every time frame, a lot of computation is required and, what is more important, the transformation parameters can change significantly within a few time frames. Especially at the beginning of the utterance when the first speech frames come in after the initial silence the quantiles can change suddenly. If the update of the transformation parameters is not restricted this will cause distortions, because the transformation will then change the signal faster than the actual signal itself changes. Then usually many insertion errors occur and error rates are higher than baseline.

The temporal change of the transformation parameters has to be slow, compared to the temporal behavior of the signal. This can be achieved by searching the updated parameter values within a small range around the previous ones $\alpha_k[t-1] \pm \delta$ and $\gamma_k[t-1] \pm \delta$, with δ in the order of $0.005 \ldots 0.01$ The changes induced by transformation the signal will then be slower than the signal's changes yielding better recognition results. As positive side effect the computational load is reduced significantly. If the step size for the grid search is also set to δ only 9 combinations of α_k and γ_k have to be evaluated, instead of the 20000 a full grid search of $\alpha_k \in [0, 1]$ and $\gamma \in [1, 3]$ would require.

If no prior information about the sentence to come is available the initial values in the first time frame should be unbiased i.e. $\alpha_k = 0$ and $\gamma_k = 1$ which corresponds to no transformation. While the sentence carrys on the transformation will adapt to the current noise conditions like the example in the figures 5.15 and 5.16 shows. There a delay of 1 second and a total window length of 5 seconds was used.

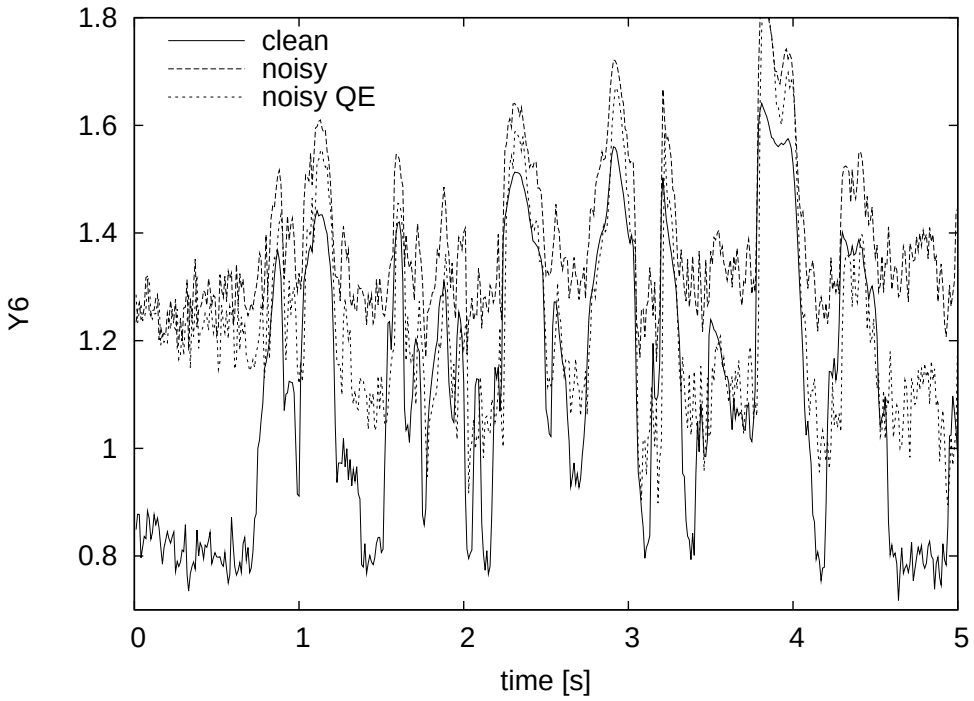

Figure 5.15: Example: output of the 6th Mel scaled filter over time for a sentence from the Aurora 4 test set before and after applying online quantile equalization with 1s delay and 5s window length.

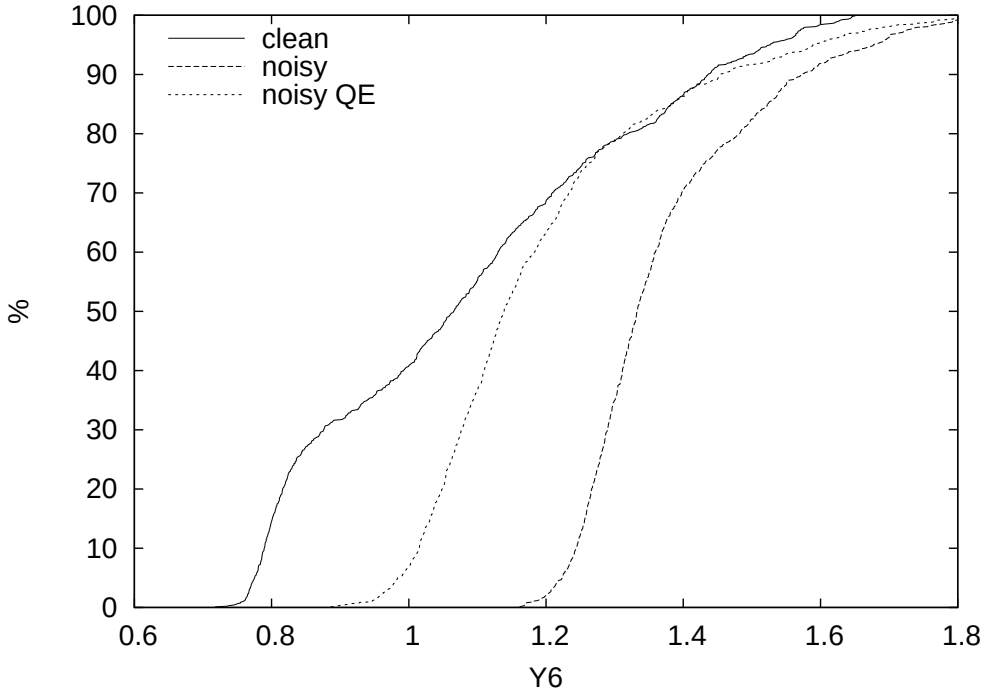

Figure 5.16: Cumulative distributions of the signals shown in Figure 5.11.

5.5 Combination of Filter–Bank Channels

Quantile equalization is applied on the filter–bank, which consists of triangular overlapping filters that are equally spaced on the Mel frequency axis. In the previous section the transformation was restricted to functions that were just applied to each individual filter, even though the output of neighboring filters is highly correlated. The dependencies between the filters should be taken into account: if a certain filter channel is more affected by noise than its neighbors it is likely that enhancing the influence of these neighbors on the channel can improve the output.

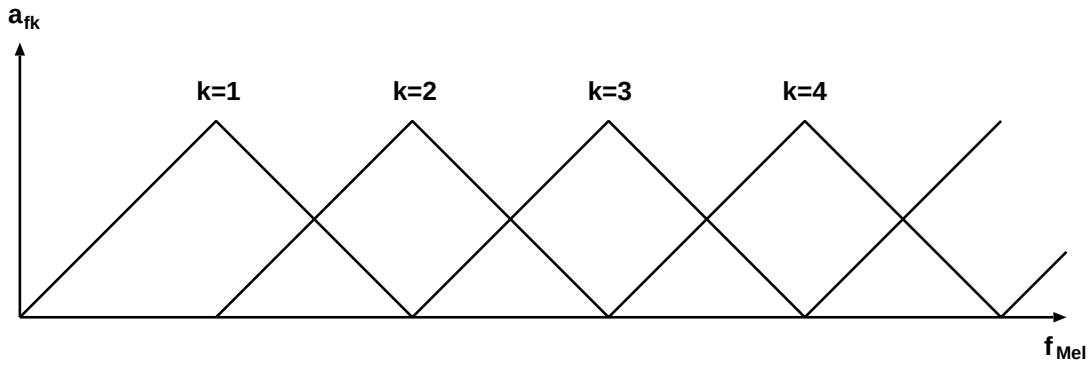

Figure 5.17: Overlapping filter–bank channels equally spaced on the Mel–frequency axis.

The problem can also be considered from the more general point of view of distributions: an individual transformation of vector components is strong restriction to the general concept of making two distributions match and there is no reason to keep it up. An individual transformation of feature space dimensions can significantly reduce the mismatch between two distributions, but even if it is non–linear it will not be able to, for instance, reverse a rotation of the feature space. So it is likely that taking into account interdependencies between the filter channels can improve the recognition performance.

This assumption was confirmed in [Molau et al. 2002]. After full histogram normalization based on some minutes of testing data, the scatter matrix of the data was calculated and a rotation applied to make its principal component match the one of the training data. The approach successfully reduced the recognition error rates [Molau et al. 2002, Molau et al. 2003], but it has the obvious disadvantage of requiring sufficiently large amount of data to reliably estimate the scatter matrix and its principal component. A moving window of a few seconds will not be able to provide enough data for such an estimation, but there is an alternative that will already work with a small amount of data: the combination of neighboring filter channels [Hilger et al. 2003, Hilger and Ney 2003]

The parametric power function transformation (equation 5.7) applied as first transformation step will not make the training and recognition quantiles match perfectly, so in a second step a linear combination of a filter with its left and right neighbor can be used to further reduce the remaining difference. Here $\tilde{Y}$ and $\tilde{Q}_{ki}$ are the filter output values and the recognition quantiles after the preceding power function transformation. The combination factors are denoted λ_k for the left neighbors and ρ_k for the right neighbors.

With $\tilde{\theta}_k = \{\lambda_k, \rho_k\}$ the second transformation step can be written as:

$$\hat{Y}_k = \tilde{T}_k(\tilde{Y}_k, \tilde{\theta}_k) = (1 - \lambda_k - \rho_k)\tilde{Y}_k + \lambda_k \tilde{Y}_{k-1} + \rho_k \tilde{Y}_{k+1} \tag{5.9}$$

This transformation corresponds to a multiplication of the filter–bank output with a tri–diagonal matrix i.e. a matrix that only has entries in the principal diagonal and the neighboring diagonals. According to the definition in equation 5.9 the sum over the entries in one row of the matrix is 1 and using $d_k = 1 - \lambda_k - \rho_k$ it can be written as:

$$\tilde{T}(\tilde{Y}, \tilde{\theta}) = \begin{pmatrix} d_1 & \rho_1 & 0 & 0 & & \dots & 0 \\ \lambda_2 & d_2 & \rho_2 & 0 & & \dots & 0 \\ 0 & \lambda_2 & d_2 & \rho_2 & & \dots & \\ 0 & \dots & & \ddots & & & \vdots \\ \vdots & \dots & & & \ddots & & 0 \\ 0 & \dots & & 0 & \lambda_{N-1} & d_{N-1} & \rho_{N-1} \\ 0 & \dots & & & 0 & \lambda_N & d_N \end{pmatrix} \begin{pmatrix} \tilde{Y}_1 \\ \tilde{Y}_2 \\ \\ \vdots \\ \\ \tilde{Y}_{N-1} \\ \tilde{Y}_N \end{pmatrix} \tag{5.10}$$

Like the transformation factors of the power function α_k and γ_k the linear combination factors λ_k and ρ_k are chosen to minimize the squared distance between the training and the transformed recognition quantiles.

$$\tilde{\theta}_k = \underset{\tilde{\theta}'_k}{\operatorname{argmin}} \left(\beta \left(\lambda_k^2 + \rho_k^2 \right) + \sum_{i=1}^{N_Q - 1} \left(\tilde{T}_k(\tilde{Q}_{ki}, \tilde{\theta}'_k) - Q_i^{train} \right)^2 \right) \tag{5.11}$$

The expression $\beta \left(\lambda_k^2 + \rho_k^2 \right)$ is a factor that penalizes large values of λ_k and ρ_k [Hastie et al. 2001]. If arbitrary values are allowed, expression $1 - \lambda_k - \rho_k$ can become negative, the signal will be distorted considerably and the recognition performance will deteriorate. To get an improvement the difference between the transformation matrix that is applied and the identical transformation should remain small.

This can be achieved by limiting λ_k and ρ_k to a fixed small range e.g. 0 to 0.1, but in general better results are obtained when a penalty factor like the one used above. A factor of typically β in the order of $0.02 \dots 0.05$ will usually limit λ_k and ρ_k to values smaller than 0.1, but higher values can occur if a large difference between the test quantiles and the training quantiles makes them necessary [Hilger et al. 2003, Hilger and Ney 2003].

Even though this restricted transformation will only lead to minor changes in some filter channels, these little transformations do make a difference in the recognition results. Especially when the noise, like car noise, is limited to certain filter channels the improvement usually is significant [Hilger et al. 2003, Hilger and Ney 2003]. In section 6.3.1 a detailed investigation on the effect of different noise types will be presented.

Like the individual transformation the combination of filter channels can be integrated into the online framework described in section 5.4. After determining the optimal individual transformation, the parameters of the combination are optimized. Here the grid search for the new values of λ_k and ρ_k is also restricted to a small range (0.005) around the

previous values to avoid sudden changes of the transformation function. Then, the two transformation steps are applied to all vectors in the current window, before calculating the new mean and subtracting it.

5.6 Summary: Quantile Equalization Algorithm

All considerations about quantile based histogram equalization made in this chapter can be summarized as follows:

- **Position of quantile equalization in the feature extraction:**
  1. In principle quantile equalization can be applied at any stage of the feature extraction if an appropriate transformation function is used.
  2. Within this work quantile equalization is applied to the output of the Mel–scaled filter–bank after compressing the dynamic range of the signal with a 10th root (figure 5.10 on page 47).
  3. It is combined with a joint mean normalization step in the way that is depicted in figure 5.14.
- **Training:**
  1. Estimate four quantiles $N_Q = 4$ on each sentence of the training database.
  2. Calculate the average quantiles Q_{ki}^{train} over all utterances.
  3. Optional: scale up the largest quantile with a factor > 1 e.g. 1.2 to allow some transformation of the highest values in recognition.
  4. Optional: pool the quantiles to get filter channel k independent reference quantiles Q_i^{train}
  5. Optional: transform the training data in a second pass according to the recognition procedure described next, before actually training the system.
- **Recognition:**
  1. Estimate filter channel specific quantiles Q_i on the test utterance.
  2. If $Q_{ki} < Q_i^{train}$ then $Q_{ki} = Q_i^{train}$
  3. Transform individual filter channels:
     - Carry out a grid search ($\theta_k = \{\alpha_k, \gamma_k\}$) with $\alpha_k \in [0, 1]$ and $\gamma_k \in [1, max]$ to minimize the squared distance between the transformed recognition quantiles and the training quantiles:

$$\theta_k = \underset{\theta_k'}{\operatorname{argmin}} \left(\sum_{i=1}^{N_Q-1} \left(T_k(Q_{ki}, \theta_k') - Q_i^{train} \right)^2 \right) \tag{5.12}$$

     - Apply the power function to the filter–bank outputs Y_k to get the transformed values:

$$\tilde{Y_k} = T_k(Y_k, \theta_k) = Q_{kN_Q} \left(\alpha_k \left(\frac{Y_k}{Q_{kN_Q}} \right)^{\gamma_k} + (1 - \alpha_k) \frac{Y_k}{Q_{kN_Q}} \right) \tag{5.13}$$

  4. Combine neighboring filter channels:

– Carry out a grid search ($\tilde{\theta}_k = \{\lambda_k, \rho_k\}$) to minimize the squared distance between the combined recognition quantiles and the training quantiles, $\beta \approx 0.02 \ldots 0.05$ is a penalty factor that keeps λ_k and ρ_k small:

$$\tilde{\theta}_k = \underset{\tilde{\theta}'_k}{\operatorname{argmin}} \left(\sum_{i=1}^{N_Q - 1} \left(\tilde{T}_k(\tilde{Q}_{ki}, \tilde{\theta}'_k) - Q_i^{train} \right)^2 + \beta \left(\lambda_k^2 + \rho_k^2 \right) \right) \tag{5.14}$$

– Apply the combination to $\tilde{Y}_k$ to get the final transformed values $\hat{Y}$:

$$\hat{Y}_k = \tilde{T}_k(\tilde{Y}_k, \tilde{\theta}_k) = (1 - \lambda_k - \rho_k)\tilde{Y}_k + \lambda_k \tilde{Y}_{k-1} + \rho_k \tilde{Y}_{k+1} \tag{5.15}$$

- **Online implementation:**

1. Initialize the transformation factors with $\alpha_k = 0$, $\gamma_k = 1$, $\lambda_k = 0$ and $\rho_k = 0$ which corresponds to no transformation.
2. Calculate the signal's quantiles Q_{ki} for a filter channel in a window around the current time frame 5.14, the window size and delay can be chosen according to the demands of the application.
3. Determine the optimal transformation factors within a small range in the order of $0.005 \ldots 0.01$ around the previous values.
4. Apply the transformation to all vectors in the current window.
5. Calculate the mean values of the resulting vectors within the window.
6. Subtract the mean to get the final vector of filter–bank coefficients.

Chapter 6

Experimental Evaluations

6.1 Introduction

In this chapter the practical aspects of quantile based histogram equalization are considered. The different properties of the algorithm that were described in the previous sections will be investigated in recognition tests and discussed in more detail. The experiments have been carried out on several databases, with different levels of complexity, to examine weather the conclusions drawn from the experiments generalize or if there are any task specific characteristics that have to taken into account. Among others, the three databases that shall be considered primarily are:

- **Car Navigation:** isolated German words recorded in cars (city and highway traffic) [Hilger and Ney 2001, Molau et al. 2003]. The recognizer vocabulary consists of 2100 equally probable words. The training data is mismatched, it was recorded in a quiet office environment.
- **Aurora 3 – SpeechDat Car:** continuous digit strings recorded in cars. Four languages are available: Danish, Finnish, German, and Spanish [Lindberg 2001, Nokia 2000, Netsch 2001, Macho 2000]. Several training and test sets with different amounts of mismatch are defined.
- **Aurora 4 – noisy WSJ 5k:** utterances read from the Wall Street Journal with various artificially added noises. The recognizer vocabulary consists of 5000 words. Two training and 14 test sets with different noises and microphone channels are defined [Hirsch 2002, Parihar and Picone 2002]).

The actual description of the experimental evaluations is divided into three main parts:

- **6.2 Baseline Results:** this section introduces the characteristics of the databases (more details can be found in appendix A), together with a description of the baseline recognizer setups and the corresponding recognition results. It also includes considerations about the improved baseline results that can be obtained when replacing the logarithm by root functions.

- **6.3 Quantile Equalization: Standard Setup:** this section presents the results that can be obtained when using quantile based histogram equalization in the recommended setup that was summarized on page 56. These results will be compared to those obtained on the same databases with other approaches.
- **6.4 Quantile Equalization: Alternative Setups:** how the recognition results change when modifying the setup of the quantile equalization is investigated in this section.

Finally, the conclusions drawn from these investigations will be presented in a summary which will finish the chapter.

6.2 Baseline Results

The following section will describe the properties the three databases that shall be considered primarily. The e baseline feature extraction and recognizer settings will be listed, before presenting the corresponding recognition results.

6.2.1 Database Definitions and Baseline Results

Isolated Word Car Navigation Database

Database Definitions:

The Car Navigation database consists of isolated German words. The 19 hours of training data were recorded in a quiet office environment, with the microphone at 30cm distance directly in front of the speaker. All recordings contain at least half a second of silence before and after the spoken word. The sampling rate was 16kHz. The detailed database statistics are shown in table A.1, appendix A.

The three test data sets of approximately 100min each where recorded in the matched office conditions (average SNR 21dB) and in real noisy car environment (city– and highway traffic, average SNRs 9dB and 6dB). The speaker sitting in the passenger seat and the microphone mounted above the speaker on the visor. The objective was to record realistic data without explicitly waiting for stationary background noise conditions, so the city traffic test sets consists of many recordings that were made during acceleration, deceleration, gear shifts and changes of the road surface.

On this database acoustic modelling can be evaluated without the influence of a language model. The words that are to be recognized are equally probable, there are 2100 words in the vocabulary.

Feature Extraction:

The baseline MFCC feature extraction implementation of the RWTH system was used [Welling et al. 1997]. It includes cepstral mean normalization.

- 25ms Hamming window, 10ms frame shift
- 1024–point fast Fourier transform
- 20 Mel scaled filter–bank channels
- 16 cepstral coefficients
- cepstral mean normalization (utterance wise if not stated otherwise)
- dynamic features: 16 first derivatives + 1 second derivative
- linear discriminant analysis [Welling et al. 1997]:

  3×33 cepstral coefficients with derivatives $\rightarrow$ 33 dimensional feature vector

Recognizer setup:

The RWTH large vocabulary continuous speech recognition system [Ney et al. 1998, Sixtus et al. 2000] was modified for the recognition of isolated words. All parameters were optimized on the clean office test set.

- triphone models
- phonetic classification and regression tree [Beulen et al. 1997] with 700 tied states
- gender independent modelling
- 21k Gaussian densities
- pooled diagonal covariance matrix
- 2100 word recognizer vocabulary
- single word recognition
- no language model

Baseline Results:

In matched clean office conditions the cepstral mean normalization is not is not needed (table 6.1). The word error rate without any normalization is 2.8%, while cepstral mean normalization 2.9%, a little difference that is negligible.

The influence of the mean normalization becomes obvious in the mismatched car noise conditions were the error rates are significantly lower than those obtained without normalization. However, the final result with cepstral mean normalization, 31.6% on the city data and 74.2% on the highway data can not be considered satisfactory, especially if these numbers are compared to the clean baseline result. The results of all further investigations will be compared to the baseline with cepstral mean normalization.

Table 6.1: Baseline results on the German isolated word Car Navigation database. LOG: logarithm, no norm.: no normalization applied, CMN: cepstral mean normalization.

Car Navigation Baseline Results			
	Word Error Rates [%]		
test set SNR [dB]	office 21	city 9	highway 6
LOG no norm.	2.8	68.0	99.0
LOG CMN	2.9	31.6	74.2

Aurora 3 SpeechDat Car

Database Definitions:

The Aurora 3 database consists of the digit string subsets from the SpeechDat–Car databases in Danish, Finnish, German, and Spanish [Lindberg 2001, Nokia 2000, Netsch 2001, Macho 2000]. The data was recorded in real car environment in a broad range of driving conditions with a close talking and a far field microphone, the sampling rate was 8kHz. In each of the language three evaluation conditions: well matched, medium, mismatch and high mismatch are defined by partitioning the data into different training and test data subsets. There are considerable differences in the amount of data that is available in the subsets of each language. The average is about 3 hours per training set and 50 minutes per test set, the exact numbers are shown in tables A.3 to A.6 in the appendix.

Reference Feature Extraction:

A Mel cepstrum feature extraction front end without any kind of normalization "Aurora WI007" [ETSI 2000, Hirsch and Pearce 2000] is the reference for all evaluations on the different Aurora databases:

- 25ms Hamming window, 10ms frame shift
- 512–point fast Fourier transform
- 23 Mel scaled filter–bank channels
- logarithmic frame energy + 12 cepstral coefficients (without the 0th)
- dynamic features: 13 first derivatives + 13 second derivative
- → 39 dimensional feature vector

Reference Recognizer Setup:

The reference recognizer for Aurora 3 digit string recognition experiments is the HTK speech recognition toolkit [Young et al. 2000], with the setup described in [Hirsch and Pearce 2000].

- HTK speech recognizer (Aurora evaluation settings [Hirsch and Pearce 2000])
- word models of the same length (16 states) for the digits
- gender independent modelling
- 552 Gaussian densities
- 11 digit recognizer vocabulary
- no language model

Aurora 3 Reference Baseline Results:

The baseline recognition results on the Aurora 3 test data are shown in table 6.2. Even in the well matched condition the average word error rate of 8.9% is high for a digit recognition task. It can be explained by the small amount of training data that is available and the acoustic modelling. The whole word models have the same same number of states of all digits in all languages, the real length of the individual digits is not taken into account. The extent of the mismatch clearly influences the error rates. In the medium mismatch case it is 22.0% word error rate and it increases to 48.9% in the high mismatch case.

The average results for the individual languages differ considerably, but a conclusion on whether digit strings utterances corrupted by noise can be recognized easier in certain languages can not be drawn from the table. The recording conditions and amount of training data that is available differs too much.

According to the official evaluation scheme the average word error rates in the last row of the table are weighted averages. To account for the differences in the baseline error rate of the tree conditions, the average relative improvement is also calculated as weighted average over the numbers for the individual conditions. Thus, even if two setups lead to the same weighted average word error rate, the corresponding relative improvement as defined by the official evaluation scheme can be different.

Table 6.2: Reference baseline result without any normalization on the Aurora 3 SpeechDat Car database. WM: well matched, MM: medium mismatch, HM: high mismatch.

	Word Error Rates [%]					rel. impr. [%]
	Finnish	Spanish	German	Danish	average	
WM × 0.40	7.3	7.1	8.8	12.7	**8.9**	0.0
MM × 0.35	19.5	16.7	18.9	32.7	**22.0**	0.0
HM × 0.25	59.5	48.5	26.8	60.6	**48.9**	0.0
average	25.6	20.8	16.9	31.7	**23.5**	0.0

Aurora 4 noisy Wall Street Journal 5k

Database Definitions:

The Aurora 4 data is based on the Wall Street Journal 5k (WSJ) database that was originally used in the ARPA evaluations. Different noise samples at various SNRs were added to turn the original data recorded in a quiet studio, into a noisy database [Hirsch 2002]. Two training sets and two sampling rates (8kHz and 16kHz) are available. The clean training data corresponds to that used in the original WSJ evaluations: 7138 utterances, corresponding to 15 hours of data, read from the Wall Street Journal, recorded with a Sennheiser HMD414 microphone. The total amount of data for the multicondition training set is identical, but noises with an average SNR of 15dB were added to 75% of the recordings and in 50% of the cases the recordings from a second microphone (one of various different types) were used. Details are summarized in table A.7 on page 128.

Table 6.3: Added noise and microphone used for the 14 test sets of the Aurora 4 database.

noise	microphone	
	Sennh.	2nd
clean	1	8
car	2	9
babble	3	10
restaurant	4	11
street	5	12
airport	6	13
train	7	14

There is a total of 14 test sets. For each of the two microphone channels there are one clean and six noisy test sets. After filtering the data with a frequency characteristic that is typical for telecommunications applications [Hirsch 2002] the noises were added at equally distributed SNRs between 15dB and 5dB. Only 166 utterances of the 330 in the original WSJ test set are used to reduce the processing time required for the 14 recognition tests [Parihar and Picone 2002]. Table A.7 summarizes the corpus statistics and table 6.3 shows how the test set numbers used later are related to the noises and microphones.

If not stated otherwise the unsegmented original test data was used for the experiments. If the data was segmented 200ms of silence were left before and after each utterance.

The recognition tests were carried out with two different recognition systems, the official reference system [Parihar and Picone 2002] and the RWTH large vocabulary speech recognition system [Ney et al. 1998, Sixtus et al. 2000].

Reference Feature Extraction:

The reference front–end for the Aurora 4 evaluations was the same baseline MFCC without any normalization that was also used for the experiments on the Aurora digit string databases [ETSI 2000, Hirsch and Pearce 2000].

Reference Recognizer setup:

The reference recognition system that sets the baseline for the Aurora 4 evaluations was provided by Mississippi State University's Institute for Signal Processing (ISIP) [Deshmukh et al. 1999]. The setup that is defined in [Parihar and Picone 2002] was used for the baseline experiments and left unchanged during the tests with quantile equalization described in section 6.3.3:

- ISIP speech recognition system (Aurora evaluation setup [Parihar and Picone 2002])
- Gender independent models
- 3215 context dependent triphone states tied using decision trees
- 12.9k Gaussian mixture densities
- Across word modeling
- Bigram language model

Aurora 4 Reference Baseline Results:

The recognition results for this setup are shown in table 6.5 on page 69 (ISIP baseline). The baseline result on the clean test set is 14.9% word error rate if the system is trained on clean data. All error rates on the noisy conditions are above 60%, because the feature extraction does not include any kind of normalization. The average over all conditions is 69.8%. Training on noisy multicondition data increases the error rate on the clean test set to 23.5%, but it leads to better recognition results on the noisy data, reducing the overall average to 39.6%.

The real–time requirements of the baseline system are considerable (page 26 of [Parihar and Picone 2002]). Therefore this system was only used for the baseline tests and one direct comparison presented in section 6.3.3, were quantile equalization was added to the given system. The more detailed investigations on different aspects of quantile equalization were carried out with the RWTH large vocabulary speech recognition system [Ney et al. 1998, Sixtus et al. 2000].

RWTH Feature Extraction:

The baseline feature extraction implementation of the RWTH system corresponds to the one shown on the left side of figure 5.10. Cepstral mean normalization is considered to be a standard method that should already be included when defining a baseline.

- 25ms Hamming window, 10ms frame shift
- 1024–point fast Fourier transform
- 20 Mel scaled filter–bank channels
- 16 cepstral coefficients

- cepstral mean normalization (utterance wise if not stated otherwise)
- linear discriminant analysis [Welling et al. 1997] (instead of derivatives):

  7×16 cepstral coefficients $\rightarrow$ 32 dimensional feature vector

RWTH Recognizer Setup:

The setup of the RWTH system was optimized on the clean test set. The intention was to investigate how the system that yields the minimal error rate on the clean data performs on the noisy test sets without readjusting any of the parameters. Table 6.4 shows the optimization of the number of densities, which was found to be optimal in the order of 200k .

- across–word triphone models [Kanthak et al. 2000b, Sixtus 2003]
- phonetic classification and regression tree [Beulen et al. 1997] with 4001 tied states
- gender independent modelling
- 210k–240k Gaussian densities
- pooled diagonal covariance matrix
- 5k recognizer vocabulary
- trigram language model

Table 6.4: Optimization of the RWTH baseline system for Aurora 4 on the clean data (test set 1). DEL: deletions, INS: insertions, SUB: substitutions, WER: word error rate.

no. of dns.	DEL [%]	INS [%]	SUB [%]	WER [%]
4.0k	1.9	1.2	7.3	10.4
7.5k	1.2	1.1	4.9	7.3
14.5k	1.1	0.8	4.0	6.0
28.5k	1.1	0.6	3.5	5.1
56.0k	1.1	0.7	2.9	4.6
104k	1.0	0.7	3.0	4.7
170k	1.1	0.6	2.9	4.6
236k	1.1	0.7	2.8	**4.5**
286k	1.1	0.8	2.7	4.6

Aurora 4 RWTH Baseline Results:

The baseline word error rate for the system trained on clean data is 4.5% on clean test data and the overall average over all the test sets is 45.7% (RWTH baseline in table 6.5). Training on noisy data considerably improves the average result, the error rate is reduced to 19.5%, but again the cost is an increase of the error rate to 8.3% on the clean subset.

The real–time requirements of the recognition were reduced with a bigram language–model look–ahead [Ortmanns et al. 1996] and fast acoustic likelihood calculation using SIMD (single instruction multiple data) instructions [Kanthak et al. 2000a]. The resulting real time factor with this setup was about 2 on the clean test set and around 10 for the noisy data sets (1800MHz AMD Athlon). A more aggressive pruning could have reduced this number below 1 on the clean test set without affecting the result, but the performance on the noisy data would have suffered too much.

A direct comparison between the ISIP and RWTH baseline results is not appropriate because the two recognizer setups differ considerably. The number of Gaussian densities used is lies in different orders of magnitude and the ISIP system uses a bigram language model, while a trigram is applied in the RWTH system.

The two baseline results will only serve as reference for the comparison of the different feature extraction front–ends, while using either of the recognizers as back–end. In accordance with the philosophy of the distributed speech recognition scenario [Pearce 2000] that is considered in the Aurora evaluations, neither the recognizer's setup nor the training procedure was readjusted when modifying the feature extraction.

Table 6.5: Baseline results on the unsegmented 16kHz Aurora 4 data. The official reference system (ISIP) does not use any normalization, the RWTH baseline that already includes cepstral mean normalization.

Aurora 4 Baseline Results															
clean training	Word Error Rates [%]														
	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	14.9	65.2	69.2	63.1	72.3	69.4	73.2	61.3	81.7	82.5	75.4	83.8	81.0	84.1	**69.8**
RWTH baseline	4.5	12.5	45.3	51.6	51.7	36.2	54.8	23.1	35.4	62.3	66.0	71.8	55.3	69.8	**45.7**

multicondition training	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	23.5	21.9	29.2	34.9	33.7	33.0	35.3	49.3	45.2	49.2	48.8	51.7	49.9	49.0	**39.6**
RWTH baseline	8.3	6.8	13.9	17.5	17.9	12.8	17.9	14.1	17.1	28.0	31.2	31.2	24.7	31.2	**19.5**

6.2.2 Comparison of Logarithm and Root Functions

The investigation presented in this section shall show, how simply replacing the logarithm in the baseline feature extraction by a root function of the form x^r can already significantly improve the recognition performance on noisy test data. Note that no specific parameter adjustments were carried out after replacing the logarithm.

Car Navigation: table 6.6 lists the results on the isolated word Car Navigation database. A window of 1s length with 500ms delay was used for the mean normalization. On the noisy city and highway test sets applying root functions clearly reduces the error rates. Applying the 20th root $r = 0.05$ already significantly improves the results, the 10th root $r = 0.1$ performs even better, and finally the minimal error rates are obtained when applying a 5th root i.e. a factor of $r = 0.2$. On the city test set the reduction from 31.6% to 17.4% word error rate corresponds to a relative improvement of 45% — on the highway data the relative improvement is even larger with 61%, or expressed as absolute numbers a reduction from 74.2% word error rate down to 28.7%. However, increasing r even further to $r = 0.5$, which corresponds to the use of a square root, does not improve the results any further, on the contrary they are deteriorated considerably.

Table 6.6: Comparison of the logarithm in the feature extraction with different root functions on the Car Navigation database. LOG: logarithm, CMN: cepstral mean normalization, 2nd – 20th: root instead of logarithm, FMN: filter mean normalization.

test set SNR [dB]		Word Error Rates [%]		
		office 21	city 9	highway 6
LOG	CMN	2.9	31.6	74.2
LOG	FMN	3.5	27.0	63.9
20th	FMN	**2.6**	22.0	49.2
10th	FMN	2.8	19.9	40.1
5th	FMN	3.3	**17.4**	**28.7**
2nd	FMN	16.7	40.8	70.0

On the clean office data the situation is different, here the minimal error rate of 2.6% is already obtained when applying the 20th root. The result of the 10th root 2.9% is comparable to the baseline with logarithm of 2.8%, while for the 5th root it increases to 3.3%. The square root again severely affects the recognition performance, apparently it does not reduce the dynamic range of the data well enough.

The results presented later in section 6.4.6 will show that quantile equalization reduces the differences between the 5th, 10th and 20th root an the noisy data sets. The conclusion that can be drawn is: the 10th root is a good compromise for systems that are to be applied in clean and noisy conditions. While there is no negative effect on the clean baseline and the result on the noisy data is already significantly improved.

Aurora 3 SpeechDat Car: the use of 10th root also improves the results on the Aurora 3 SpeechDat Car databases in a similar way. The detailed overview for all languages and mismatch conditions are presented in table 6.7. The upper part of the table (LOG) shows the reference baseline without any normalization. The improvement through mean

normalization with only 10ms delay and 5s window length is shown in the middle part (LOG FMN). Replacing the logarithm by the 10th root (10th FMN) only has a little influence in the well matched condition, the error rate just decreases from 9.7% to 9.0%. In the medium mismatch condition the error rate reduction is larger again (from 19.9% to 17.4%) and it increases even more in the high mismatch case (from 34.0% to 22.7%). So, in the overall weighted average the 10th root can contribute to a reduction of the word error rate from 19.3% to 15.4% on these databases.

Table 6.7: Comparison of logarithm and 10th root. Detailed recognition results on the Aurora 3 SpeechDat Car databases. rel. impr.: relative improvement over the reference baseline setup (page 63) without any normalization. WM: well matched, MM: medium mismatch, HM: high mismatch.

		Word Error Rates [%]					rel. impr.
		Finnish	Spanish	German	Danish	average	[%]
LOG	WM × 0.40	7.3	7.1	8.8	12.7	**8.9**	0.0
	MM × 0.35	19.5	16.7	18.9	32.7	**22.0**	0.0
	HM × 0.25	59.5	48.5	26.8	60.6	**48.9**	0.0
	average	25.6	20.8	16.9	31.7	**23.5**	0.0
LOG FMN	WM × 0.40	7.7	6.3	7.7	17.0	**9.7**	-3.9
	MM × 0.35	20.0	10.7	18.5	30.6	**19.9**	10.5
	HM × 0.25	34.5	26.0	20.1	55.4	**34.0**	30.6
	average	18.7	12.7	14.6	31.3	**19.3**	9.8
10th FMN	WM × 0.40	4.9	8.2	8.3	14.8	**9.0**	1.6
	MM × 0.35	12.5	11.2	17.8	28.0	**17.4**	22.3
	HM × 0.25	26.1	17.5	17.7	29.6	**22.7**	51.4
	average	12.9	11.6	13.9	23.1	**15.4**	21.3

Aurora 4 noisy WSJ: the Wall Street Journal database with artificially added noises offers the possibility of investigating the effect of the logarithm and different root functions in more detail by directly comparing corresponding clean and noisy test sets.

When training the system on clean data (upper part of table 6.8) the results have a tendency that is similar to the one that was already observed on the Car Navigation database. The word error rate on the clean test data (set no. 1) is similar for the logarithm, 20th and 10th root (about 4.5%). It rises when applying the 5th root (5.1%) and the square root deteriorates the result (8.9%). The average over all noise conditions is again minimal for the 5th root, from an initial baseline of 45.7% for the logarithm it is reduced to 24.5% which corresponds to a relative reduction of 46%.

When training the system on noisy multicondition data the tendency is different (lower part of table 6.8). Here the 5th root leads to the best result on the clean test set 1. From a baseline of 8.3% the error rate is reduced down to 6.3%, but this advantage does not pay off in the overall average. The average result for 5th, 10th and 20th root is similar. The error rate for the 10th root is 17.8%, compared to the baseline of 19.5% this is a relative improvement of 9%.

Table 6.8: Comparison of the logarithm in the feature extraction with different root functions on the Aurora 4 noisy WSJ 16kHz database. LOG: logarithm, CMN: cepstral mean normalization, 2nd – 20th: root instead of logarithm, FMN: filter–bank mean normalization.

clean training		Word Error Rates [%]														
		test set														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP	baseline	14.9	65.2	69.2	63.1	72.3	69.4	73.2	61.3	81.7	82.5	75.4	83.8	81.0	84.1	**69.8**
LOG	CMN	4.5	12.5	45.3	51.6	51.7	36.2	54.8	23.1	35.4	62.3	66.0	71.8	55.3	69.8	**45.7**
20th	FMN	4.5	9.3	24.6	34.4	34.2	25.4	34.3	22.7	30.1	44.1	51.9	54.3	45.2	50.4	**33.2**
10th	FMN	4.4	8.6	20.8	27.5	28.7	20.9	30.1	22.3	28.9	41.9	45.3	48.6	41.4	46.3	**29.7**
5th	FMN	5.1	8.3	15.1	19.3	18.8	17.9	18.8	22.8	21.6	37.1	38.6	40.3	39.5	39.6	**24.5**
2nd	FMN	8.9	19.1	18.0	20.3	21.5	18.8	21.9	34.9	44.0	43.5	45.1	46.1	43.6	43.5	**30.7**

multicondition training		test set														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP	baseline	23.5	21.9	29.2	34.9	33.7	33.0	35.3	49.3	45.2	49.2	48.8	51.7	49.9	49.0	**39.6**
LOG	CMN	8.3	6.8	13.9	17.5	17.9	12.8	17.9	14.1	17.1	28.0	31.2	31.2	24.7	31.2	**19.5**
20th	FMN	7.8	6.9	12.1	15.1	16.2	12.1	16.6	13.7	15.1	25.4	28.4	29.5	24.6	29.2	**18.0**
10th	FMN	7.2	6.9	12.4	15.1	16.3	12.2	15.4	14.4	16.7	24.8	27.5	28.3	23.1	28.3	**17.8**
5th	FMN	6.3	7.7	12.2	14.3	15.7	11.6	15.5	15.6	18.6	25.9	26.8	28.8	24.3	28.2	**18.0**
2nd	FMN	11.1	13.4	17.0	19.0	19.5	17.2	18.4	28.8	32.9	37.3	39.4	39	37.3	37.6	**26.3**

In conclusion the results on Aurora 4 again confirm that the 10th root seems to be a good compromise that works well in all training and test conditions, so in the following it will be used as standard. The 5th root is well suited for systems that shall only recognize noisy data while being trained on clean data, but this advantage is lost when the system is trained on multicondition data.

Table 6.9: Correlation (equation 6.1) between the clean and noisy test data sets of the Aurora 4 database compared to the average word error rates and the corresponding error rate on the clean subset (clean training data).

test set	Correlation				
	LOG	20th	10th	5th	2nd
1	1.00	1.00	1.00	1.00	1.00
2	0.73	0.76	0.78	0.83	0.92
3	0.71	0.74	0.76	0.81	0.91
4	0.70	0.72	0.74	0.78	0.88
5	0.67	0.70	0.72	0.77	0.88
6	0.71	0.74	0.76	0.80	0.90
7	0.65	0.68	0.71	0.76	0.87
8	0.74	0.76	0.78	0.82	0.87
9	0.61	0.64	0.66	0.71	0.80
10	0.59	0.62	0.64	0.69	0.78
11	0.60	0.61	0.63	0.67	0.75
12	0.55	0.57	0.59	0.64	0.74
13	0.60	0.62	0.65	0.69	0.78
14	0.56	0.58	0.61	0.65	0.75
average	0.67	0.70	0.72	0.76	0.84
average WER [%]	45.7	33.2	29.7	24.5	30.7
clean set 1 WER [%]	4.5	4.5	4.4	5.1	8.9

A possible explanation why the root functions are able to outperform the logarithm can be found when calculating the correlation between the original clean and noisy signals (Y^{clean} and Y^{noisy}) after the logarithm respectively the different root functions. Here the correlation coefficient *cor* calculated over all filter channels $k \in 1, \ldots, K$ and time frames $t \in 1, \ldots, T$ of the test set is defined as:

$$cor = \frac{\mathrm{cov}\left(Y_k^{clean}[t], Y_k^{noisy}[t]\right)}{\sqrt{\mathrm{var}\left(Y_k^{clean}[t]\right)\,\mathrm{var}\left(Y_k^{noisy}[t]\right)}} \tag{6.1}$$

cov denotes the covariance and var the variance.

Obviously the result is $cov = 1$ for the clean data and the correlation decreases with growing mismatch. While the overall average is 0.67 for the logarithm, it is 0.72 when using 10th root and 0.76 when the 5th root is applied (table 6.9). Of course one can not expect that this increase of the correlation between the clean data set and the noisy data sets is directly related to the reduction of the average word error rate of the recognition.

The results for the 2nd root illustrate that: the initial error rate on the clean test set 1 is very high, so in this case the large correlation between that data set and the noisy ones is no indication for good overall recognition results. Nonetheless the correlation can still be considered to be a measure for the mismatch between the clean and noisy data, so if the initial error rate on the clean data set is low and the average correlation is high, the average error rate is likely to be low too.

The figures 6.1 to 6.4 on pages 75 and 76 visualize how the difference in correlation influences different representations of the data. The examples on page 75 show the signals for an individual sentence over time, the example is the one that was already used in the introduction to the previous chapter. The signals after logarithm are shown in figure 6.1 and after the 10th root in 6.2). The two plots seem very similar, the only apparent difference is the clean signal's variability in the non–speech portion at the beginning of the utterance, which is smaller when the 10th root is applied. This little difference results in a small difference of the correlations. The correlation on the sentence is 0.80 when the logarithm is used and 0.84 in the 10th root case.

To illustrate the effect of the different compression functions on the entire test set the representation as scatter plot is given in the figures 6.3 and 6.4. Each point in the plots represents one time frame of the test data, the current amplitude of the noisy signal is plotted against the corresponding original clean amplitude. If the clean and noisy signal were identical the result would be the diagonal line in this representation. A point below the diagonal indicates that the amplitude of the noisy signal is higher than the one of the original clean signal and vice versa.

In both cases the typical effect that was already discussed in the introduction to chapter 5 can be observed again: the low amplitude non–speech parts of the signal are more affected by the noise than the higher amplitude speech parts. This causes a significant shift of the points corresponding to silence away from the diagonal, while the speech portions are not distorted that much and the points still scatter around the diagonal.

Besides the additive noise the microphone mismatch in the example also shifts the points away from the diagonal. The distinct structure of two parallel speech regions with high density is caused by the different frequency characteristics of the recordings with the mismatched microphones [Hirsch 2002]. Figure 6.5 shows the influence of the microphone mismatch alone, without additional noise.

The range of values in which the data points lie is different in the figures 6.3 and 6.4. To make them comparable the vertical range is scaled to the approximately the same hight. This reveals that the range of values of the noisy signal, i.e. the horizontal extension of the cloud of points is smaller when applying the logarithm (figure 6.3). Which makes the classification more difficult. When applying the 10th root the horizontal range gets larger (6.4), which reduces the confusability of the points. Expressed in terms of correlation: it is 0.65 for the logarithmic plot and 0.69 for the 10th root plot.

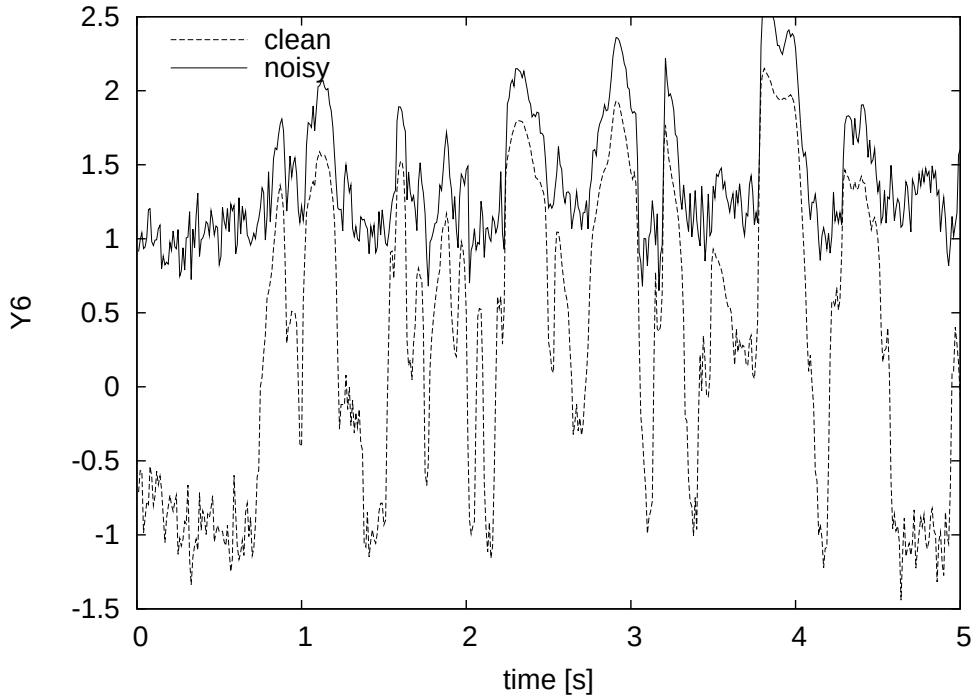

Figure 6.1: Output of the 6th Mel scaled filter after logarithm over time for a sentence from the Aurora 4 test set, clean data and noisy data with additive street noise and microphone mismatch. The correlation of the two signals is 0.80.

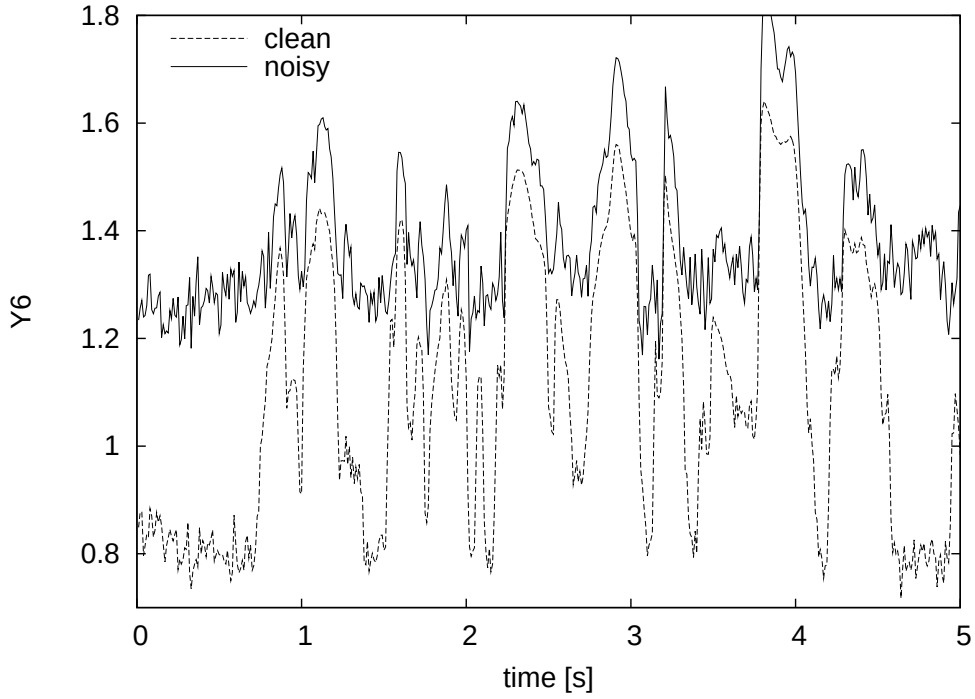

Figure 6.2: Example: output of the 6th Mel scaled filter after 10th root over time for a sentence from the Aurora 4 test set, clean data and noisy data with additive street noise and microphone mismatch. The correlation of the two signals is 0.84.

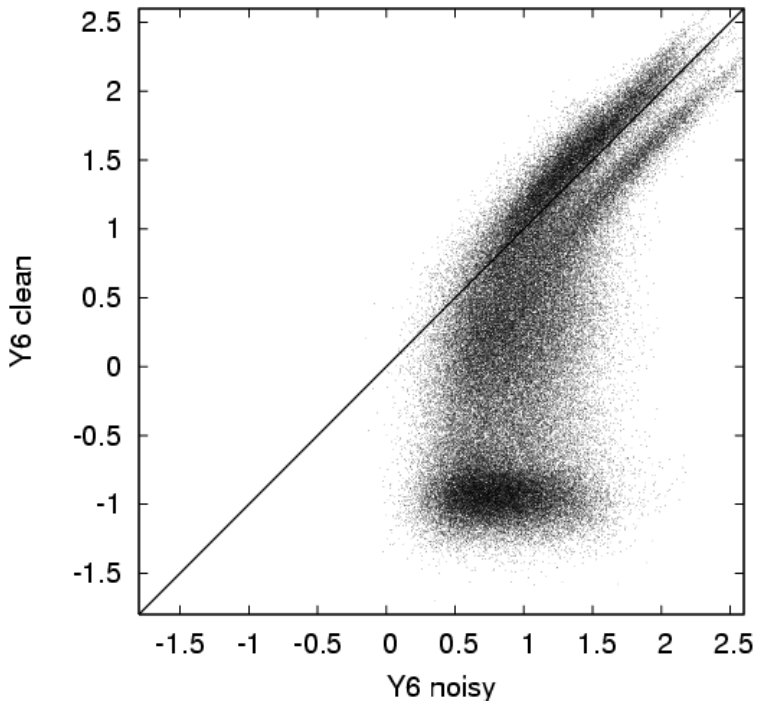

Figure 6.3: Scatter plot clean data vs. noisy data on the filter–bank (additive street noise and microphone mismatch) after applying a logarithm. The correlation of this set of points is 0.65.

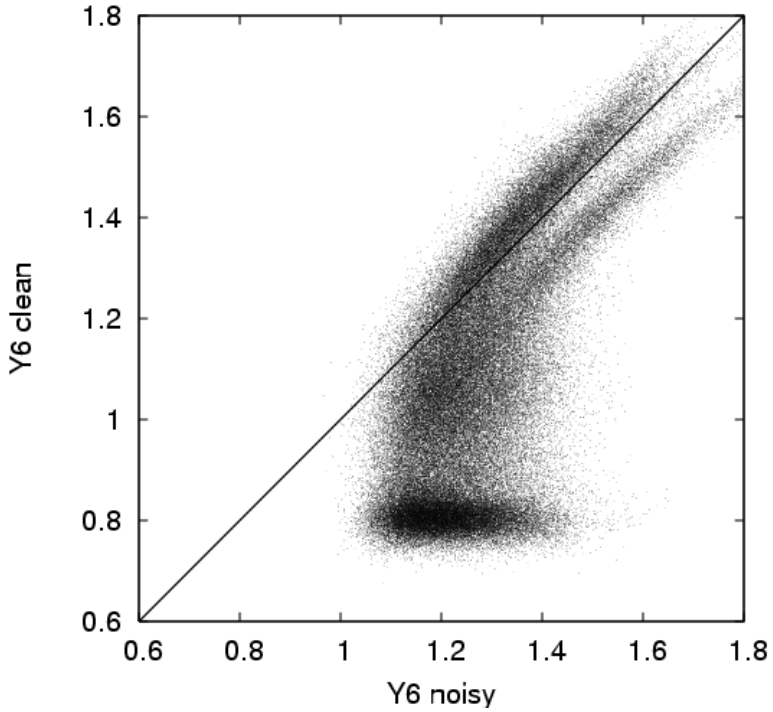

Figure 6.4: Scatter plot clean data vs. noisy data on the filter–bank (additive street noise and microphone mismatch) after applying the 10th root. The correlation of this set of points is 0.69.

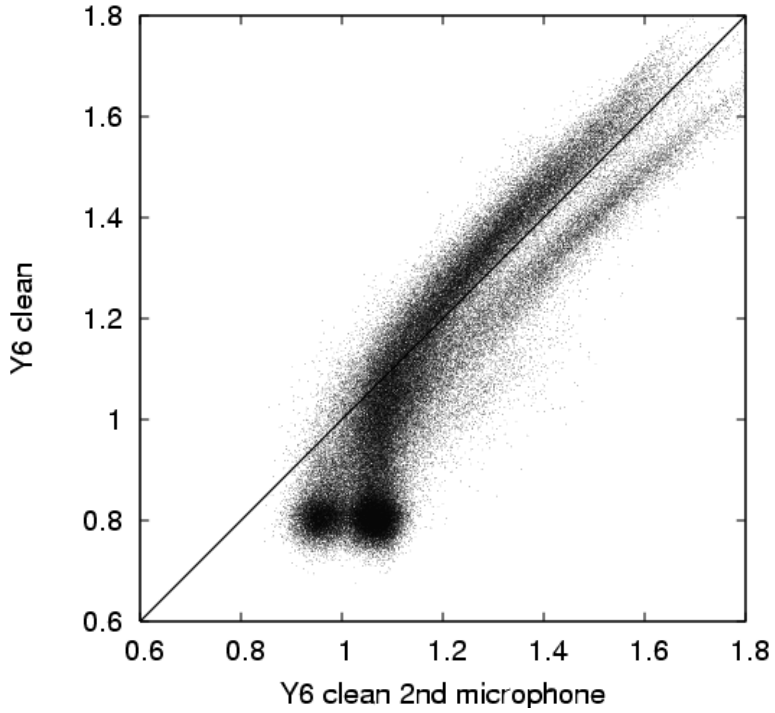

Figure 6.5: Scatter plot clean data vs. data with microphone mismatch after applying the 10th root. Three different microphones were used in the recordings [Hirsch 2002], but only one of them has a significantly different influence in the considered filter channel.

Additional tests: a particularly large influence of the 10th root was also observed in additional tests with a voice user interface database recorded by Lucent Bell Labs [Afify and Siohan 2001]. The recognition task consisted of digit strings and command phrases recorded in cars. The training data was recorded with a close talking microphone. The test data is mismatched, the microphone was mounted on the visor. The baseline recognition result with the recognizer setup described in [Afify and Siohan 2001] was 20.2%. Without changing the recognizer setup, by simply replacing the logarithm in the front–end by a 10th root the error rate could be be halved to 10.2% (cf. section 6.3.1 and [Hilger et al. 2003]).

Conclusions:

- Simply replacing the logarithm in the feature extraction by a root function of the form x^r consistently improves the recognition of noisy data. r should be in the range of [0.05, 0.2].
- The 5th root, $r = 0.2$ works especially well if a system shall recognize noisy data while being trained on clean data, this advantage is lost when the system is trained on multicondition data.
- The 10th root $r = 0.1$ is a good compromise that works well in all training and test conditions.
- 10th root will be used as standard for all following experiments.

6.2.3 10th Root, Mean and Variance Normalization

The combination of mean and variance normalization shall be considered in this section. Especially when the mismatch between the training and test data is high variance normalization can lead to considerable improvements (table 6.10).

Car Navigation: the dramatic increase of word error rate on the clean test set of the Car Navigation database can to some extent be explained by the moving window implementation that was used. Mean and variance normalization were carried out in a window of 1s length with 500ms delay. The typical recording in the Car Navigation database has more than 500ms of silence before the utterance, so the variance normalization will initially only normalize the variance of the silence. When the first speech frames come in the variance will change suddenly leading to distortions. But even if the mean and variance normalization are carried out utterance wise there still is an increase of the error rate on the clean test data, the corresponding results then are: clean 4.8%, city 16.3%, and highway 24.1%. A similar result was reported in [Molau et al. 2002] were cepstral mean and variance normalization were used in a standard feature extraction with logarithm.

Table 6.10: Car Navigation database: influence of variance normalization. LOG: logarithm, CMN: cepstral mean normalization, 10th: root instead of logarithm, FM(V)N: filter mean (and variance) normalization.

	Word Error Rates [%]		
test set SNR [dB]	office 21	city 9	highway 6
LOG CMN	2.9	31.6	74.2
LOG FMN	3.5	27.0	63.9
10th FMN	2.8	19.9	40.1
10th FMVN	8.5	15.1	24.1

Aurora 4 noisy WSJ: on the Aurora 4 database no improvement through variance normalization could be obtained (table 6.11) even though it was applied utterance wise. A dramatic increase of insertion errors occurred. To counteract that effect the original approach of optimizing the language model scaling factor on the clean data and using this factor on all noisy test sets was given up for the these tests, but even the readjustment of the language model scale was not able to prevent the increase of insertion errors. The final recognition results for joint mean and variance normalization (FMVN) are still higher than the previous ones with mean normalization (FMN).

Table 6.11: Average recognition results on the Aurora 4 noisy WSJ 5k database. LOG: logarithm, CMN: cepstral mean normalization, 10th: root instead of logarithm, FM(V)N: filter mean (and variance) normalization.

	Word Error Rates [%]	
	clean training	multi. training
LOG CMN	45.7	19.5
10th FMN	29.7	17.8
10th FMVN	33.8	19.8

Conclusions:

- Variance normalization is not able to provide consistent improvements that are independent from the database.
- On the Car Navigation Database variance normalization leads to significant improvements when the mismatch between the training and the test data is important, but the recognition on clean test data suffers.
- Variance normalization induces a considerable increase of deletion errors on the Aurora 4 database that can not be compensated.
- In the following tests variance normalization will not be applied.

6.3 Quantile Equalization: Standard Setup

6.3.1 Recognition Results with the Standard Setup

In this section recognition results for quantile equalization after 10th root compression are presented. Quantile equalization was used in the baseline setup described in the summary on page 56, while the original baseline recognizer setup (page 67) was left unaltered again. Considerations about alternative setups will follow in the final part of the chapter.

Car Navigation: on the Car Navigation database with equally probable isolated words the effect of quantile equalization is particularly large. The feature extraction setup with 10th root and filter mean normalization was used as starting point. Mean normalization and quantile equalization were again applied with a moving window implementation that had 500ms delay and 1s window length. The results are shown in table 6.12.

Quantile equalization clearly outperforms simple variance normalization discussed in the previous section (table 6.10). When applying the transformation of individual filter–bank channels the error rates are reduced from 19.9% to 11.7% on the city test set and from 40.1% to 20.1% on the highway test set (10th QE FMN), this corresponds to a relative improvement of over 70% compared to the original baseline setup with the logarithm. The result on the clean test set suffers somewhat from quantile equalization, the error rate is increased from 2.8% to 3.2%. Since quantile equalization aims at reducing the mismatch between current test data and the training data it can only have a contribution if there is a mismatch, this is what these numbers confirm.

Table 6.12: Recognition results on the Car Navigation database with quantile equalization applied only during the recognition tests. LOG: logarithm, CMN: cepstral mean normalization, 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF(2): quantile equalization with filter combination (2 neighbors).

test set SNR [dB]			Word Error Rates [%]		
			office 21	city 9	highway 6
LOG		CMN	2.9	31.6	74.2
10th		FMN	2.8	19.9	40.1
10th	QE	FMN	3.2	11.7	20.1
10th	QEF	FMN	3.6	10.3	17.1
10th	QEF2	FMN	3.6	9.6	17.1

Car noise primarily affects the low filter channels, so a combination of neighboring filter channel can be expected to improve the recognition and this expectation is met. Considering the direct left and right neighbors leads to an improvement on the noisy test sets (10th QEF FMN in table 6.12). But again the price for this improvement is an increase of the error rate on the clean condition. Taking two neighbors into account (10th QEF2 FMN) can reduce the result on the city data even further. The final word error rate is 9.6%. The other conditions remain unchanged. Going an other step further and taking into account three neighbors does not yield any improvements.

Table 6.13: Recognition results for the Aurora 4 noisy WSJ 16kHz databases. LOG: logarithm, CMN: cepstral mean normalization, 10th: 10th root instead of logarithm, FMN: filter–bank mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination. Utterance wise mean normalization and quantile equalization.

clean training	Word Error Rates [%]														
	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	14.9	65.2	69.2	63.1	72.3	69.4	73.2	61.3	81.7	82.5	75.4	83.8	81.0	84.1	**69.8**
LOG CMN	4.5	12.5	45.3	51.6	51.7	36.2	54.8	23.1	35.4	62.3	66.0	71.8	55.3	69.8	**45.7**
10th FMN	4.4	8.6	20.8	27.5	28.7	20.9	30.1	22.3	28.9	41.9	45.3	48.6	41.4	46.3	**29.7**
10th QE FMN	4.3	8.3	16.2	21.1	22.1	19.7	22.8	22.0	27.8	37.2	40.7	41.1	39.0	40.5	**25.9**
10th QEF FMN	4.2	7.7	15.7	21.3	21.6	18.9	23.1	21.2	26.6	36.7	40.3	40.7	38.4	40.1	**25.5**
	relative improvement over LOG CMN [%]														
10th FMN	2.2	31.2	54.1	46.7	44.5	42.3	45.1	3.5	18.4	32.7	31.4	32.3	25.1	33.7	**35.0**
10th QE FMN	4.4	33.6	64.2	59.1	57.3	45.6	58.4	4.8	21.5	40.3	38.3	42.8	29.5	42.0	**43.3**
10th QEF FMN	6.7	38.4	65.3	58.7	58.2	47.8	57.8	8.2	24.9	41.1	38.9	43.3	30.6	42.6	**44.2**

multicondition training	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	23.5	21.9	29.2	34.9	33.7	33.0	35.3	49.3	45.2	49.2	48.8	51.7	49.9	49.0	**39.6**
LOG CMN	8.3	6.8	13.9	17.5	17.9	12.8	17.9	14.1	17.1	28.0	31.2	31.2	24.7	31.2	**19.5**
10th FMN	7.2	6.9	12.4	15.1	16.3	12.2	15.4	14.4	16.7	24.8	27.5	28.3	23.1	28.3	**17.8**
10th QE FMN	7.1	7.1	11.3	14.7	14.6	11.6	14.1	14.3	16.9	24.2	27.5	27.0	23.1	25.6	**17.1**
10th QEF FMN	7.0	6.9	11.5	14.9	14.5	11.6	14.1	14.4	16.9	23.8	27.4	27.0	22.8	25.7	**17.0**
	relative improvement over LOG CMN [%]														
10th FMN	13.3	-1.4	10.8	13.7	8.9	4.7	14.0	-2.0	2.3	11.4	11.9	9.3	6.5	9.3	**8.7**
10th QE FMN	14.5	-4.3	18.7	16.0	18.4	9.4	21.2	-1.3	1.2	13.6	11.9	13.5	6.5	17.9	**12.3**
10th QEF FMN	15.7	-1.4	17.3	14.9	19.0	9.4	21.2	-2.0	1.2	15.0	12.2	13.5	7.7	17.6	**12.8**

Aurora 4 noisy WSJ: On the Aurora 4 database the influence of different training conditions and different types of noises can be investigated. The system with 10th root and utterance wise filter mean normalization is again used as starting point for the investigations, (10th FMN in table 6.13 on page 82). When training on the original clean data quantile equalization can reduce the average error rate from 29.7% to 25.9% (10th QE FMN) and the combination of neighboring filter channels yields an other small improvement to 25.5% (10th QEF FMN). So in the end the overall improvement through quantile equalization is not as large as on the Car Navigation database. On that database the result was only determined by the acoustics of the isolated words. The situation on the Aurora 4 database is different, here more side effects influence the result: the recognition task consists of continuous speech, across–word models are used, the acoustic model complexity is larger and a language model (trigram) is involved.

Looking at the results for the individual noise conditions in the second part of table 6.13 shows that in this case there is no loss when applying quantile equalization on the clean data (set no. 1). Quantile equalization leads to larger relative improvements on the data sets without microphone mismatch (conditions 1–7). Apparently the transformation function that was designed to reduce the influence of noise by scaling down the signal is not suited that well for mismatched microphone conditions.

The combination of neighboring filters has the largest contribution in the clean training condition on the car data (set no. 2), where it can increase the relative improvement from 33.6% to 38.4%.

In the setup with multicondition training data the mismatch between the test and training is much smaller, so the contribution of quantile equalization can be expected to be lower and the results also confirm that. The error rate with 10th root and mean normalization was 17.8% (10th FMN), it is only to 17.0% (10th QEF FMN). The tendency that the improvement on the test sets 1–7 without microphone mismatch is higher is still there.

Putting the contributions of the 10th root, quantile equalization with filter combination and mean normalization together — and comparing that result to the original baseline with logarithm and mean normalization (LOG CMN) leads to the following results: the average error rate is reduced from rate from 45.7% to 25.5% (i.e. 44.2% relative) when training on clean data, and from 19.5% to 17.0% (i.e. 12.8% relative) in the multicondition training case. Like on the Car Navigation database quantile equalization has a larger effect when the mismatch is bigger.

The influence of quantile equalization on the data can be visualized very well in the scatter plot representation. In the original plot after the 10th root (figure 6.6) the region corresponding to silence is shifted away from the diagonal, the points scatter over a large area and there are two distinct speech regions corresponding to the different microphones. As figure 6.7 shows, applying quantile equalization changes the scatter plot considerably. The silence region is shifted back towards the diagonal and it becomes more compact. The two speech regions are merged, so the overall cloud of points is more dense too and the number points below and above the diagonal is almost balanced after the transformation.

Calculating the correlation over the entire test data sets according to equation 6.1 reveals an increase from 0.72 to 0.78 by quantile equalization. The combination of neigh-

Table 6.14: Correlation (equation 6.1) between the clean and noisy test data sets of the Aurora 4 database compared to the average word error rates and the corresponding error rate on the clean subset (clean training data).

	Correlation			
test set	LOG	10th	10th QE	10th QEF
1	1.00	1.00	0.99	0.99
2	0.73	0.78	0.85	0.86
3	0.71	0.76	0.80	0.80
4	0.70	0.74	0.78	0.78
5	0.67	0.72	0.76	0.77
6	0.71	0.76	0.81	0.81
7	0.65	0.71	0.74	0.75
8	0.74	0.78	0.82	0.82
9	0.61	0.66	0.77	0.78
10	0.59	0.64	0.73	0.73
11	0.60	0.63	0.71	0.71
12	0.55	0.59	0.69	0.70
13	0.60	0.65	0.73	0.74
14	0.56	0.61	0.69	0.70
average	0.67	0.72	0.78	0.78

boring filters does not have any significant impact on the correlation, there are slight differences in the different test sets, but the overall average remains the same.

The contribution of quantile equalization with filter combination depends on the number of densities that are used in the acoustic model. If single densities (4k in table 6.15) or the large number of more than 50 densities per sate (220k) are used, the relative improvement is somewhat smaller than in the other cases (14–100k).

Table 6.15: Recognition results on the Aurora 4 data with different numbers of densities.

	Word Error Rates [%]									
	clean training					multicondition training				
	4.0k	14.5k	28.5k	100k	220k	4.0k	14.5k	28.5k	100k	220k
10th FMN	45.1	37.4	35.7	32.3	29.7	32.5	24.2	21.7	18.9	17.8
10th QEF FMN	38.6	30.4	28.7	25.6	25.5	31.0	22.7	19.7	17.9	17.0

	relative improvement over 10th FMN [%]									
10th QEF FMN	16.8	18.7	19.6	26.2	14.1	4.8	6.2	9.2	5.6	4.5

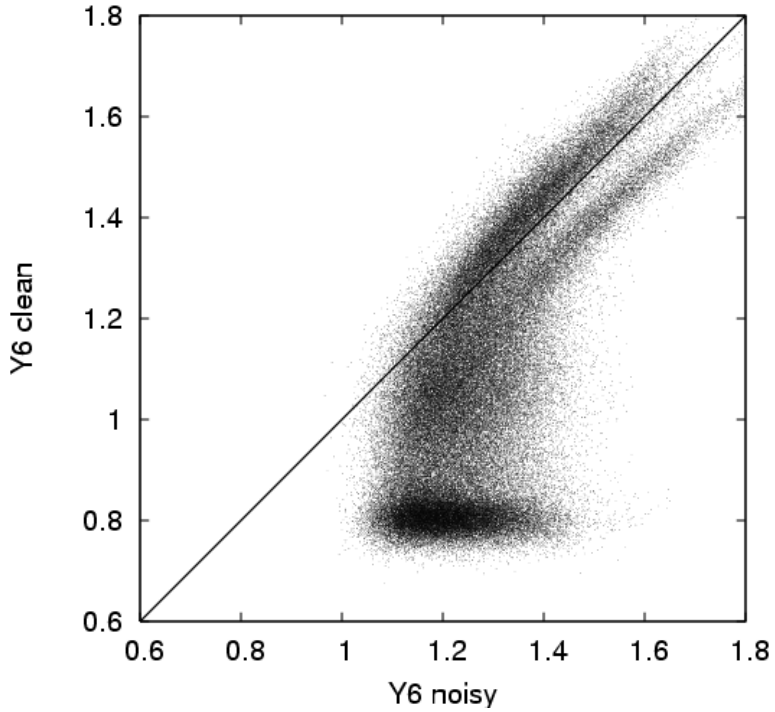

Figure 6.6: Scatter plot clean data vs. noisy data on the filter–bank (additive street noise and microphone mismatch) after applying the 10th root. The correlation of the set of points is 0.69.

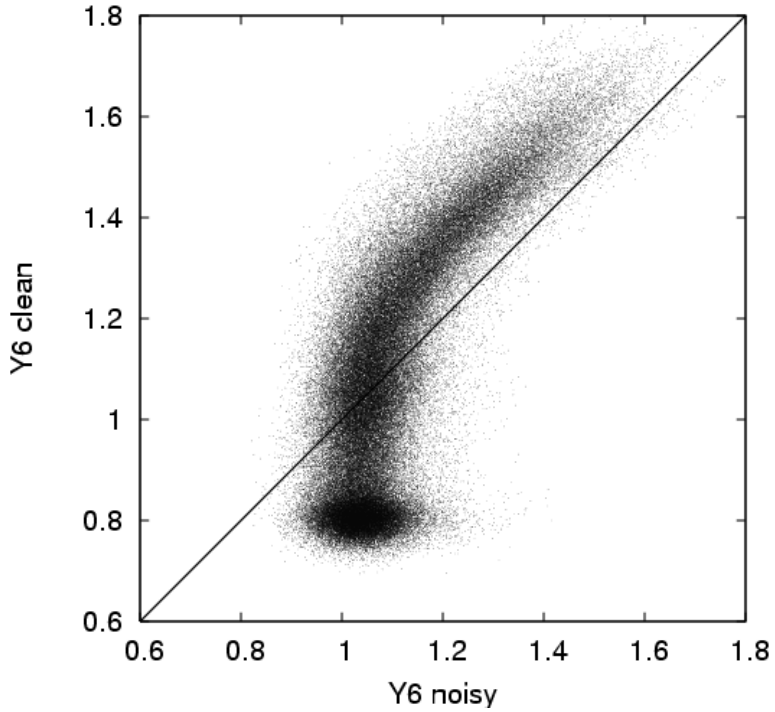

Figure 6.7: Scatter plot clean data vs. noisy data on the filter–bank (additive street noise and microphone mismatch) after applying the 10th root and quantile equalization. The correlation of the set of points is 0.77.

Additional tests: Some additional tests [Hilger et al. 2003] on other databases, with the Lucent Bell Labs recognition system as back–end, also underline the genericity of the quantile equalization setup described here:

The 9h of training data of the **car voice user interface (VUI) database** were recorded in cars using a close talking microphone (different driving conditions, some recordings with background music), the test data (30min, digit strings and command phrases) with a microphone mounted on the visor. The Lucent Bell Labs speech recognition system with the setup described in [Afify and Siohan 2001] was used for the recognition test, in which the digits and 85 different command words were modeled with triphones. A finite state grammar determined the allowed command phrases.

Table 6.16: Recognition result on the car VUI database. LOG: logarithm, CMN: cepstral mean normalization, 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination.

			Word Error Rates [%]
LOG		CMN	20.2
10th		CMN	10.2
10th	QE	CMN	9.5
10th	QEF	CMN	7.6

On this database the 10th root already halves the error rate, table 6.16. The further improvement of the quantile equalization (online version with 10ms delay and 5s window length) is not as large, because even though a close talking microphone was used for the recordings, the training data is somewhat noisy again, so the overall mismatch between the training and test data is smaller than it was on the Car Navigation database. However the combination of the filters can still reduce the error rate to 7.6% which is a 20% improvement over the best result with normal quantile equalization.

Table 6.17: Recognition result on a car–telephone digit string database. LOG: logarithm, CMN: cepstral mean normalization, 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination.

test set			Word Error Rates [%]			
			handset	tele_tsclr	sdn 10	lapel
LOG		CMN	0.5	0.8	2.7	3.2
10th		CMN	0.5	0.8	2.5	3.3
10th	QE	CMN	0.4	0.8	2.5	2.8
10th	QEF	CMN	0.3	0.8	2.5	2.6

If the recordings are not that seriously affected by the noise and the initial error rates are low, the improvements through quantile equalization can be expected to be even smaller [Hilger et al. 2003]. This is shown in table 6.17 which presents the results of a **digit string recognition task in car environment** [Li et al. 2001]. The Lucent Bell Labs context dependent head–body–tail digit models [Afify et al. 2001] were used.

Handset (24min), tele_tsclr (68min) and sdn10 (5h 18min) denote the test data sets with different telephone handsets, there is abundant training data (88h) that matches these test conditions. Lapel (58min) is a mismatched test set which was recorded using a lapel microphone.

The baseline error rates on this database are much lower than on the previous databases and mismatch between the training and test conditions is small, so the improvements through quantile equalization are expectedly small. On two of the telephone handsets minor improvements can be observed (handset and sdn 10 in Table 6.17). Only the mismatched lapel test set in the last column of the table shows a significant improvement from 3.2% word error rate to 2.8% using normal quantile equalization and a further improvement to 2.6% when combining the filters.

The German **EMBASSI database** ("Elektronische Multimediale Bedien- und Service Assistenz") consists of natural language command phrases to control a television set with video recorder [Haderlein and Nöth 2003]. Different microphone channels and recording conditions quiet and with background noises are available. For the investigations described in the following, the microphone at 1m was used (channel 6) in training and test. The partitioning in training and test data corresponds to the description in [Haderlein et al. 2003]. The detailed database statistics are summarized in the appendix on page 128.

The feature extraction setup corresponds to the one used for the tests on the Aurora 4 database (page 66). 3 consecutive 33 dimensional feature vectors (with 16 original cepstrum components, 16 derivatives, and 1 second derivative) are concatenated and an LDA is applied to reduce the final dimensionality to 33. For the normalization 1 second delay and 5 seconds window length were used. On the 1 hour of training data an acoustic model with 500 tied triphone states and 16.6k Gaussian densities was trained. The recognizer vocabulary consists of 474 words, a trigram language model is used.

The first column in table 6.18 shows the results on on quiet, manually segmented data. The numbers can be considered as best case lower bound of the error rate. The baseline word error rate is 4.0%, using the 10th root increases the error rate to 4.4% and quantile equalization does not help in this quiet case.

With regard to a real application real application the other columns of the table are more realistic. The audio streams were not segmented, the recognizer has to cope with continuous audio streams that consist of long silence, respectively background noise portions between the actual utterances. "Disturber" denotes a second person walking through the recording room, speaking from time to time, whose utterances should not be recognized. "Newsreader" denotes continuous background speech and "music" denotes more or less loud background music. These noises are played via loudspeaker and disturb the speaker.

The recognition word error rate of the baseline system ranges from 38.9% to 89.8% (table 6.18). The use of the 10th root consistently decreases the error rates and quantile equalization leads to a further improvement in all conditions. The combination of neighboring filter channels does not help in this type of noise conditions. Putting everything together the largest improvement is obtained on the test set with the music at a moderate volume. The error rate is reduced from a baseline of 56.5% to 46.4%.

Table 6.18: Recognition results on the EMBASSI database. The microphone was positioned in front of the speaker, at a distance of 1m. LOG: logarithm, FMN: filter mean normalization, 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination.

			Word Error Rates [%]				
test condition			quiet	disturber	disturber newsreader	disturber music	disturber loud music
segmentation			segmented	unsegmented audio streams			
LOG		FMN	4.0	38.9	58.0	56.5	89.8
10th		FMN	4.4	35.5	54.9	48.4	79.5
10th	QE	FMN	4.4	34.0	54.0	46.6	77.3
10th	QEF	FMN	4.6	34.4	53.5	46.4	77.6

These experiments show that quantile equalization is helpful on this database, but not yet sufficient to bring the error rates back down to the level of the quiet unsegmented recordings. There is room for improvement, e.g. through the use of microphone array approaches together with a reliable noise robust segmentation of the data streams.

Conclusions:

- Quantile based histogram equalization is generally applicable. On several databases with different complexity and together with various recognizers as back–end significant error rate reductions were obtained.
- The relative improvements through quantile equalization depend on the amount of mismatch between the training and test data. With increasing mismatch the relative improvements increase.
- The combination of neighboring filter channels usually leads to additional small improvements.
- If the noise primarily affects a few filter–bank channels (like car noise) the relative improvement through filter combination is bigger.

6.3.2 Comparison with Other Approaches

Full histogram normalization: the result of the comparison between quantile equalization and full histogram normalization [Molau 2003] depends on the characteristics of the data that is to be recognized.

A speaker session of the Car Navigation database contains an average of about 7 minutes of data, that can be used for full histogram normalization [Molau et al. 2001]. The SNR, especially in the city traffic sessions, is not constant over all recordings, but the differences are rather small so this is not really problematic.

The comparison between the quantile equalization of individual filter channels, that was only based on 1 second of data (line 3 in table 6.19), and the corresponding full histogram normalization (line 7) reveals that simply increasing the amount of data used to estimate the transformation does not necessarily improve the results [Molau et al. 2003]. Histogram normalization is better on the clean data, but on the noisy test sets the results are the same.

The influence of the 10th root is reduced by histogram normalization (lines 6 and 7). The setup with logarithm even is a little better on the clean and city data, so it was used for the further investigations.

The advantage of histogram normalization pays off when taking into consideration the amount of silence in the recordings, which varies between 45% and 75% in the different speaker sessions. If a first recognition pass is used to determine the amount of silence and the target histograms are adapted correspondingly [Molau et al. 2002] the error rate can be reduced on all test sets yielding results (line 8) that are lower than those obtained with quantile equalization.

Table 6.19: Comparison of quantile equalization with histogram normalization on the Car Navigation database. QE train: applied during training and recognition. HN: speaker session wise histogram normalization, HN sil: histogram normalization dependent on the amount of silence, ROT: feature space rotation.

test set SNR [dB]			Word Error Rates [%]		
			office 21	city 9	highway 6
(1) LOG		CMN	2.9	31.6	74.2
(2) 10th		FMN	2.8	19.9	40.1
(3) 10th	QE train	FMN	3.3	10.1	16.7
(4) 10th	QEF train	FMN	3.6	8.8	15.9
(5) 10th	QEF2 train	FMN	3.6	7.7	15.1
(6) 10th	HN	CMN	3.3	10.9	15.1
(7) LOG	HN	CMN	2.8	10.2	16.6
(8) LOG	HN sil	CMN	2.6	8.2	14.3
(9) LOG	HN sil ROT	CMN	2.4	7.1	11.1

The feature space rotation [Molau et al. 2002], which is also based on the 7 minutes of data per speaker available results in a further improvement of the results (line 9), even

on the clean data. As to be expected these are lower than the best results with quantile based simple filter combination (line 5). Compared to the other data sets the difference on the city traffic set is rather small 7.1% compared to 7.7%. This can be explained by the changing environment conditions in the city traffic data. As long as the noise conditions are fairly constant over a long time, like in the clean and highway test sets, full histogram normalization can take a clear advantage of the large amount of data it uses. In non–stationary conditions more data does not mean better adaptation, the advantage is reduced or completely lost.

Table 6.21 shows the results on the Aurora 4 database, with the detailed results for histogram normalization (10th HN FMN). Like in previous experiment the histograms of the test data were estimated per speaker session (about 2.6 minutes of data each). Figure 6.8 illustrates an example from the test data: histogram normalization clearly reduces the mismatch between the noisy and the clean cumulative distribution function. But this smaller mismatch does not really pay off in terms of reduced word error rate. There is a reduction by 1.2% absolute to 28.5% WER for the system trained on clean data, but there is no improvement at all in the case of multicondition training.

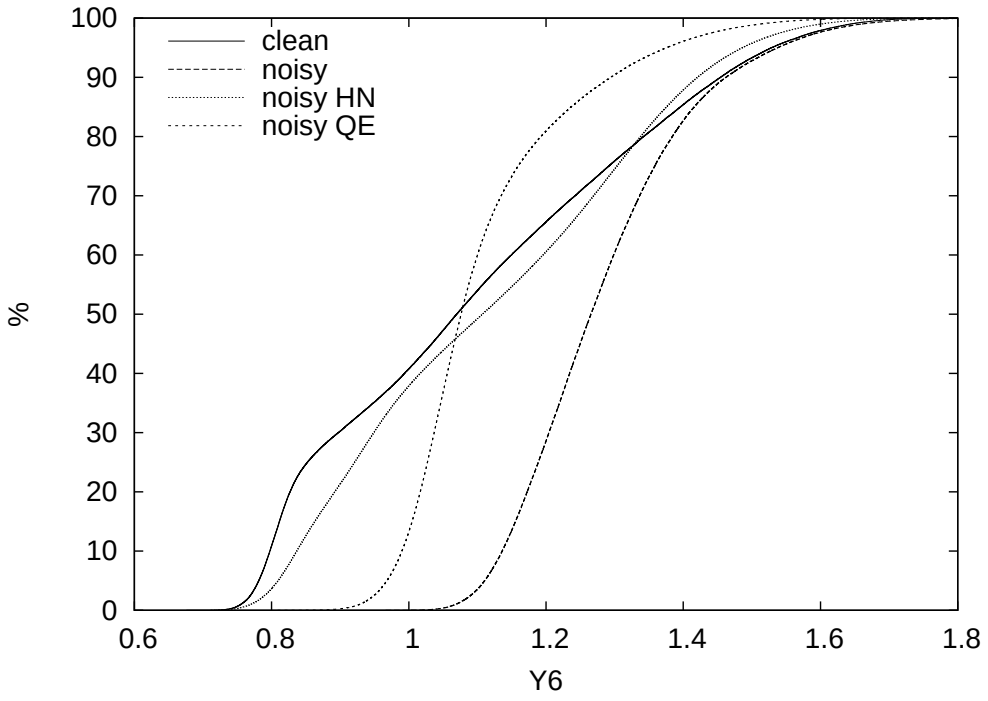

Figure 6.8: Cumulative distribution function of the 6th filter output. clean: data from test set 1, noisy: test set 12, noisy HN: after histogram normalization, noisy QE: after quantile equalization.

The characteristics of the Aurora 4 database can explain this result. On the other databases used to test histogram normalization [Molau 2003] a speaker session contained recordings with the same acoustic conditions. In the case of Aurora 4 the noises that were added to the data at different SNRs lead to significantly different conditions within one "session."

Table 6.20: Comparison of full histogram normalization HN and quantile equalization QE with the RWTH recognizer

clean training	Word Error Rates [%]													
	test set													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	14.9	65.2	69.2	63.1	72.3	69.4	73.2	61.3	81.7	82.5	75.4	83.8	81.0	84.1	**69.8**
LOG CMN	4.5	12.5	45.3	51.6	51.7	36.2	54.8	23.1	35.4	62.3	66.0	71.8	55.3	69.8	**45.7**
10th FMN	4.4	8.6	20.8	27.5	28.7	20.9	30.1	22.3	28.9	41.9	45.3	48.6	41.4	46.3	**29.7**
10th HN FMN	4.8	8.3	16.3	24.4	20.6	24.3	20.4	18.7	30.5	45.9	48.5	46.4	49.5	40.8	**28.5**
10th QE FMN	4.3	8.3	16.2	21.1	22.1	19.7	22.8	22.0	27.8	37.2	40.7	41.1	39.0	40.5	**25.9**
10th QEF FMN	4.2	7.7	15.7	21.3	21.6	18.9	23.1	21.2	26.6	36.7	40.3	40.7	38.4	40.1	**25.5**

multicondition training	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	23.5	21.9	29.2	34.9	33.7	33.0	35.3	49.3	45.2	49.2	48.8	51.7	49.9	49.0	**39.6**
LOG CMN	8.3	6.8	13.9	17.5	17.9	12.8	17.9	14.1	17.1	28.0	31.2	31.2	24.7	31.2	**19.5**
10th FMN	7.2	6.9	12.4	15.1	16.3	12.2	15.4	14.4	16.7	24.8	27.5	28.3	23.1	28.3	**17.8**
10th HN FMN	7.7	6.9	11.4	13.2	14.0	13.7	13.3	15.2	18.7	28.5	27.4	28.1	27.0	25.5	**17.9**
10th QE FMN	7.1	7.1	11.3	14.7	14.6	11.6	14.1	14.3	16.9	24.2	27.5	27.0	23.1	25.6	**17.1**
10th QEF FMN	7.0	6.9	11.5	14.9	14.5	11.6	14.1	14.4	16.9	23.8	27.4	27.0	22.8	25.7	**17.0**

Table 6.21: Comparison of the correlation (equation 6.1) after histogram normalization and quantile equalization.

test set	LOG	10th	10th HN	10th QE	10th QEF
1	1.00	1.00	0.97	0.99	0.99
2	0.73	0.78	0.84	0.85	0.86
3	0.71	0.76	0.77	0.80	0.80
4	0.70	0.74	0.74	0.78	0.78
5	0.67	0.72	0.73	0.76	0.77
6	0.71	0.76	0.78	0.81	0.81
7	0.65	0.71	0.72	0.74	0.75
8	0.74	0.78	0.87	0.82	0.82
9	0.61	0.66	0.76	0.77	0.78
10	0.59	0.64	0.68	0.73	0.73
11	0.60	0.63	0.66	0.71	0.71
12	0.55	0.59	0.64	0.69	0.70
13	0.60	0.65	0.69	0.73	0.74
14	0.56	0.61	0.65	0.69	0.70
average	0.67	0.72	0.75	0.78	0.78

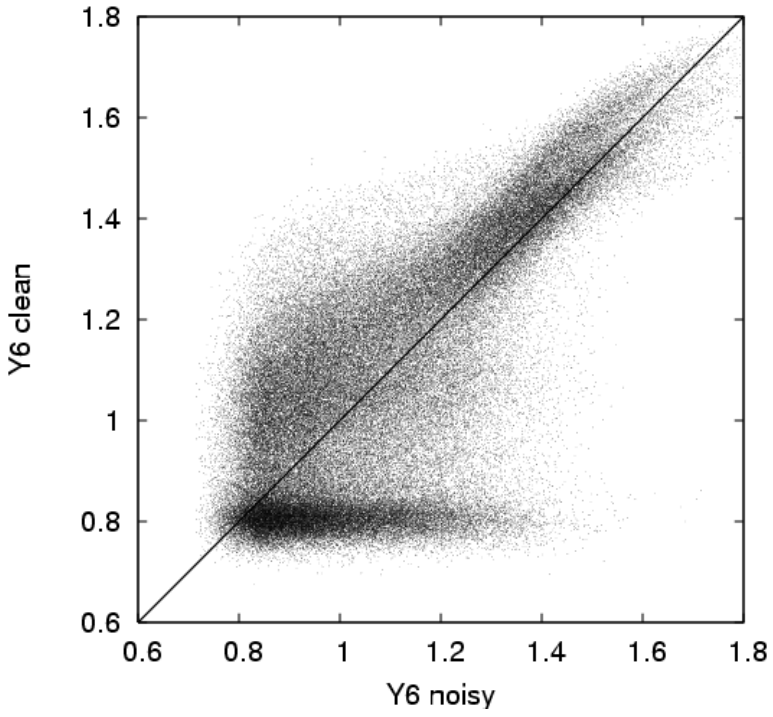

Figure 6.9: Scatter plot clean data vs. noisy data on the filter–bank (additive street noise and microphone mismatch) after applying the 10th root and histogram normalization.

In these non–stationary conditions a better matching overall cumulative distribution does not necessarily mean a there is a better match between the transformed and the original clean signal on a time–frame by time–frame level and that is what really influences the final recognition result. The correlation after histogram normalization is smaller than after quantile equalization (table 6.21). The points in the scatter plot are dispersed over a large area around the diagonal (figure 6.9). The correlation of the points shown in the example is 0.73 as compared to 0.77 when using quantile equalization (figure 6.7).

Appropriately clustering the test data by SNR and using these clusters instead of the speaker sessions might give better recognition results. For the target histogram estimated on the training data the situation is similar: while it is appropriate to estimate one target histogram for clean training data it might be better to estimate several SNR dependent target histograms on the multicondition training data. But such a procedure would be in contradiction to the goal of having a simple method that is independent from the characteristics of specific databases.

Table 6.22: Comparison of recognition results on the unsegmented Aurora 4 noisy Wall Street Journal data. The KU Leuven system [Stouten et al. 2003] uses an acoustic model with 400k Gaussian densities, the Panasonic system 32k [Rigazio et al. 2003].

		WER [%]		
sampling rate		16kHz		average
training condition		clean	multi	
ISIP Miss. State	reference baseline	69.8	39.6	**54.7**
KU Leuven	improved LDA	42.5	–	–
	improved LDA, model based enhan.	**35.7**	–	–
Panasonic	subband feature extraction	34.7	24.9	29.8
	subband feature extraction, MLLR	33.3	25.6	29.5
	MFCC, PMC	30.6	21.8	26.2
	MFCC, PMC, MMIE training	28.5	21.1	**24.8**
RWTH	28.5k densities 10th FMN	35.7	21.7	28.7
	28.5k densities 10th QEF FMN	28.7	19.7	**24.2**
	220k densities 10th FMN	29.7	17.8	23.8
	220k densities 10th QEF FMN	25.5	17.0	**21.3**

Results reported by other groups: in the following quantile equalization is compared to results reported by other research groups that also conducted experiments on the Aurora 4 database without using the reference recognition system. The numbers were published in the proceedings of the 2003 European Conference on Speech Communication and Technology in Geneva.

The Katholieke Universiteit Leuven presented a combination of an improved linear discriminant analysis algorithm, combined with a feature enhancement approach [Stouten et al. 2003], based on the combination of a clean speech model with a noise model and their relation given by a vector Taylor series approximation. [Moreno et al. 1996]. An acoustic model with 2000 tied across–word triphone states and 400k Gaussian densities was used [Stouten et al. 2003]. A bigram language model was applied. The proposed

method relies on clean speech model, so recognition results are only presented for the clean training condition (table 6.22).

The Panasonic Speech Technology Laboratory investigated several approaches individually and in combination [Rigazio et al. 2003]. The acoustic model consisted of 800-1500 tied states with 32k Gaussian densities. A wavelet based subband feature extraction combined with MLLR [Leggetter and Woodland 1995] was tested, as well as a system using a standard MFCC feature extraction together with predictive model combination [Gales 1998]. The system using the second approach was also trained in a maximum mutual information estimation (MMIE) framework. Table 6.22 lists the corresponding results. The best system yields an overall average of 24.8% word error rate. For some other approaches they also investigated, like histogram equalization, spectral subtraction and Jacobian adaptation [Rigazio et al. 2003], the authors report degradation of the overall average.

The results obtained with RWTH system using quantile equalization are competitive. The suboptimal system with a small acoustic model (28.5k densities) already yields an average word error rate of 24.2%, the final result for the optimized system with 220k densities is 21.3%.

Conclusions:

- Full histogram equalization outperforms quantile equalization if large amounts of data from stationary environment conditions are available for the estimation of the histograms.
- In non–stationary conditions, were the SNR changes from one utterance to the next or even within an utterance, quantile equalization can be recommended.
- Compared to various approaches investigated by other groups, the results obtained with quantile based histogram equalization are competitive. An overall average word error rate of 22.6% is obtained on the 16kHz Aurora 4 noisy Wall Street Journal data.

6.3.3 Results with Aurora Reference Recognizers

In this section the experimental results with the standardized reference recognition systems defined for the ETSI Aurora evaluations [Pearce 2000] will be presented. The idea behind these evaluations is to investigate the performance different of feature extraction front–ends without explicitly optimizing the setup of the back–end recognition system.

The reference recognizer for Aurora 2 and 3 digit string recognition experiments is the HTK speech recognition toolkit [Young et al. 2000], with the setup described in [Hirsch and Pearce 2000]. The baseline large vocabulary recognizer for Aurora 4 was provided by the Mississippi State University's Institute for Signal and Information Processing (ISIP) [Parihar and Picone 2002].

The initial reference feature extraction for all experiments is the so called Aurora WI007 front–end [Hirsch and Pearce 2000], a standard MFCC feature extraction without any normalization. Quantile equalization was added to this given front–end, to make sure no other side effects are introduced by e.g. using a different baseline MFCC implementation. The only modifications were: the logarithm was replaced by the 10th root, in consequence the 0th cepstral coefficient was used instead of the log–energy coefficient, and finally quantile equalization with joint mean normalization was added. The window length for used for quantile equalization and mean normalization was 5 seconds on all databases. For the Aurora 2 and 3 tests the delay was reduced to the minimum of 1 time frame i.e. 10ms, while a delay of 100 time frames (1 second) was granted for the large vocabulary tests on Aurora 4.

Aurora 2 noisy TI digits: by introducing a mean normalization and replacing the logarithm with a 10th root the overall average word error rate on the Aurora 2 noisy TI digit string database (cf. appendix on page 125 and [Hirsch and Pearce 2000]) is reduced from 27.5% to 18.2% (10th FMN in table 6.23), adding quantile equalization reduces it to 16.6% (10th QE FMN). The combination of neighboring filter channels does not lead to any significant further improvement, so the final result on Aurora 2 is 16.4% (10th QEF FMN). The combination of neighboring filter channels usually only leads to minor changes of the signal, because the typical combination coefficients are very small. Apparently, these minor transformations can only have a positive effect on the error rate if the acoustic models are better than the ones used in the standard Aurora 2 setup [Hirsch and Pearce 2000]: digit models with the same number of states for all digits and that only have three densities per state.

A more detailed look at the results again shows the dependency between the improvement and the mismatch. While the overall relative improvement in the clean training case is about 50%, it is 16% in the multicondition case. Interestingly, there even is an improvement on test set A, although the noises added to that test set correspond to the ones used in training. This shows that even in artificially added noise conditions simply providing matched noisy training data does not guarantee minimal error rates. Quantile equalization is still able to reduce the error rates by reducing the remaining mismatch between the individual test utterances and the overall average distribution of the training data.

Aurora 3 SpeechDat Car: this database [Lindberg 2001, Nokia 2000, Netsch 2001, Macho 2000] consists of real noisy recordings made in cars, but this does not change

Table 6.23: Aurora 2 noisy TI digit strings, HTK reference recognizer. rel. impr.: relative improvement over the reference baseline setup without any normalization (page 63). set A: matched noised, set B: noises not seen in training, set C: noise and frequency characteristics mismatch.

		Word Error Rates [%]				rel. impr.
		set A	set B	set C	average	[%]
LOG	multi. train.	11.9	12.8	15.4	13.0	0.0
	clean train.	41.3	46.6	34.0	41.9	0.0
	average	25.6	29.7	24.7	**27.5**	**0.0**
10th FMN	multi. train.	11.1	11.4	8.6	10.7	14.0
	clean train.	27.0	23.3	28.2	25.8	41.2
	average	19.1	17.3	18.4	**18.2**	**27.6**
10th QE FMN	multi. train.	10.2	10.8	10.8	10.5	14.9
	clean train.	23.5	21.9	22.4	22.6	49.7
	average	16.9	16.3	16.6	**16.6**	**32.3**
10th QEF FMN	multi. train.	8.9	11.4	10.9	10.3	16.2
	clean train.	23.3	21.9	22.3	22.5	49.6
	average	16.1	16.6	16.6	**16.4**	**32.9**

Table 6.24: Aurora 3 SpeechDat Car databases, HTK reference recognizer. rel. impr.: relative improvement over the reference baseline setup (page 63) without any normalization. WM: well matched, MM: medium mismatch, HM: high mismatch.

		Word Error Rates [%]					rel. impr.
		Finnish	Spanish	German	Danish	average	[%]
LOG	WM × 0.40	7.3	7.1	8.8	12.7	8.9	0.0
	MM × 0.35	19.5	16.7	18.9	32.7	22.0	0.0
	HM × 0.25	59.5	48.5	26.8	60.6	48.9	0.0
	average	25.6	20.8	16.9	31.7	**23.5**	**0.0**
10th FMN	WM × 0.40	4.9	8.2	8.3	14.8	9.0	1.6
	MM × 0.35	12.5	11.2	17.8	28.0	17.4	22.3
	HM × 0.25	26.1	17.5	17.7	29.6	22.7	51.4
	average	12.9	11.6	13.9	23.1	**15.4**	**21.3**
10th QE FMN	WM × 0.40	4.5	7.8	7.5	12.4	8.0	10.9
	MM × 0.35	12.1	10.1	16.5	23.5	15.5	29.7
	HM × 0.25	20.1	16.5	16.5	26.6	19.9	56.7
	average	11.1	10.8	12.9	19.8	**13.7**	**28.9**
10th QEF FMN	WM × 0.40	4.5	8.0	7.6	12.1	8.0	11.0
	MM × 0.35	12.2	10.1	16.8	23.4	15.6	29.1
	HM × 0.25	20.6	16.4	16.6	26.8	20.1	56.4
	average	11.3	10.8	13.1	19.7	**13.7**	**28.7**

the overall tendency of the results: 10th root and mean normalization yield a mayor reduction of the weighted average word error rate from 23.5% to 15.4% (10th FMN in table 6.24) and quantile equalization leads to 13.7% (10th QE FMN) according to the official evaluation scheme this corresponds to an average relative improvement of 29%. Again the improvement depends on the amount of mismatch and there even is an improvement in the well matched (WM) case, in which the training data covers the same broad range of noise conditions that also occur in the test data. Transforming the individual utterances of the test data to make them match the average distribution of the training data is more efficient than just providing matched training data.

These results on Aurora 2 and 3 can be compared to those presented by other groups in the Aurora special sessions of the EUROSPEECH 2001 conference in Aalborg, the ICSLP 2002 in Denver and the EUROSPEECH 2003 in Geneva.

The advanced Aurora front–end [Macho 2000, ETSI 2002] a joint development by Motorola, France Télécom, and Alcatel can be taken as reference. It includes several methods to increase the robustness: a two stage Wiener filter, SNR–dependent waveform processing, blind equalization of the cepstrum, and a voice activity detection. This complex front–end which was optimized for the Aurora 2 and 3 task outperforms quantile equalization on these databases. The overall average is 10.7% word error rate on Aurora 2 and 9.7% on Aurora 3, as compared to 16.4% and 13.7% with joint quantile equalization and mean normalization in the online setup with 10ms delay.

Compared to other single step approaches, without additional segmentation using a voice activity detection, such as spectral subtraction and histogram normalization implementations presented by other groups, the method proposed in this work yields similar results. For Aurora 2 with clean training data [Kim et al. 2003] report 26.0% and [Segura et al. 2002] report 27.7% using spectral subtraction, as compared to 22.5% with quantile equalization. Results for spectral subtraction on Aurora 3 are also reported by [Segura et al. 2002]: 17.9%, 12.4% and 11.3% for Finnish, Spanish, and German respectively, as compared to 11.3%, 10.8%, and 13.1% (cf. table 6.24).

A histogram equalization approach using a 3rd order spline function to approximate the CDF was used by [de Wet et al. 2003]. The result on Aurora 2 with clean training is 18.2% word error rate (cf. table 6.23), the result for the multicondition case was not reported. Similar results, also on Aurora 2, were presented by [de la Torre et al. 2002], using histogram equalization with a Gaussian of zero mean and unit variance as target distribution: 10.3% word error rate when using multicondition training data and 19.2% in the clean training case, while online quantile equalization yields 10.3% and 22.5% (cf. table 6.23). In a succeeding paper [Segura et al. 2002] the authors present results for the combination of spectral subtraction and histogram equalization: 9.7% and 15.7% on Aurora 2. The corresponding numbers for Aurora 3 are 13.3%, 7.0%, and 8.8% again for Finnish, Spanish, and German respectively (cf. table 6.24).

Quantile equalization outperforms spectral subtraction and, even though an online implementation that only uses 1 time–frame delay was applied here, the results are comparable to those reported by other groups [de la Torre et al. 2002, de Wet et al. 2003] that used utterance wise histogram equalization approaches.

Aurora 4 noisy WSJ: table 6.26 shows the detailed results for the Aurora 4 large vocabulary evaluations [Hirsch 2002] that were already presented in [Hilger and Ney 2003]. Both training sets were used on the 8kHz and 16kHz data respectively. The results for segmented test data are shown because the number of insertion errors considerably rises when using the reference recognizer on the original unsegmented data.

Like in all previous tests the average word error rates is significantly reduced by simply applying 10th root compression and mean normalization (10th FMN). The relative contribution of filter specific quantile equalization (10th QE FMN) and the combination of neighboring filter channels (10th QEF FMN) is smaller than in the previous tests with the RWTH system presented in Table 6.13 on page 82, but still consistent.

As to be expected from the previous tests, the largest relative error rate reductions are observed when the system is trained on clean data, the lowest absolute error rates when training on multi condition data. A look at individual noise conditions shows, quantile equalization and filter combination yield the highest error rate reductions on the test sets with the stationary band limited car noise (2 and 9).

In total, the overall average average word error rate is reduced form 48.8% to 35.6%. This final error rate is higher than the one obtained with the RWTH system (table 6.13), but it is comparable to the results reported by other groups when using the reference recognizer on the Aurora 4 database [Parihar and Picone 2003]. The Qualcomm ICSI OGI feature extraction [Benitez et al. 2001] that uses LDA derived RASTA filters [van Vuuren and Hermansky 1997] together with cepstral mean and variance normalization yielded 37.5% as overall average. For the advanced Aurora front end [Macho et al. 2002, ETSI 2002] an average of 34.5% was reported. A look at the individual conditions shows that, compared to quantile equalization, the improvement is due to better numbers for the 8kHz data.

Table 6.25: Comparison of recognition results on the Aurora 4 data using the standard reference recognizer [Parihar and Picone 2003].

		WER [%]				
sampling rate		8kHz		16kHz		average
training condition		clean	multi	clean	multi	
ISIP Miss. State: reference baseline		57.9	38.8	60.8	38.0	48.9
Qualcomm, ICSI, OGI		43.2	33.6	40.7	32.4	37.5
RWTH	10th FMN	42.3	34.2	39.4	32.7	37.2
	10th QE FMN	41.7	34.0	37.7	32.1	36.4
	10th QEF FMN	40.3	32.9	37.6	31.5	**35.6**
Motorola, France Télécom, Alcatel		37.5	31.4	37.2	31.5	34.5

Table 6.26: Standard ISIP reference recognizer back–end: results for the 8kHz and 16kHz segmented Aurora 4 noisy WSJ 16kHz databases. baseline: MFCC front end without normalization, 10th: 10th root instead of logarithm, FMN: filter–bank mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination. 1s delay 5s window length.

8kHz clean training	Word Error Rates [%]														
	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	16.2	49.6	62.2	58.7	58.2	61.5	61.7	37.4	59.7	69.8	67.7	72.2	68.3	67.9	**57.9**
10th FMN	15.8	22.3	42.0	45.4	45.7	43.9	47.7	26.2	34.0	51.7	54.2	56.5	51.8	54.9	**42.3**
10th QE FMN	15.6	22.2	41.8	45.0	43.3	44.4	46.0	25.4	33.4	51.1	53.9	56.1	51.6	53.9	**41.7**
10th QEF FMN	15.3	21.5	39.9	44.3	42.3	42.8	45.2	24.3	31.0	48.4	51.6	54.9	49.0	53.2	**40.3**
multicondition training	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	18.4	24.9	37.6	39.3	38.8	38.2	40.4	29.7	37.3	48.3	46.1	50.6	44.9	49.3	**38.8**
10th FMN	23.5	21.8	31.3	35.9	36.6	33.8	37.8	25.9	29.2	39.2	41.9	42.1	37.3	41.9	**34.2**
10th QE FMN	22.6	22.4	31.0	36.4	36.4	34.5	36.2	24.5	28.3	39.3	42.5	42.2	37.8	41.8	**34.0**
10th QEF FMN	21.4	21.4	30.3	35.5	35.0	33.0	37.0	22.9	27.6	38.2	40.2	40.8	36.0	40.8	**32.9**

16kHz clean training	Word Error Rates [%]														
	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	14.0	56.6	57.2	54.3	60.0	55.7	62.9	52.7	74.3	74.3	67.5	75.6	71.9	74.7	**60.8**
10th FMN	14.5	18.9	33.4	41.1	37.5	34.8	38.7	32.8	38.9	49.4	52.3	56.7	48.3	53.8	**39.4**
10th QE FMN	14.0	19.1	31.6	38.0	34.5	33.0	37.3	31.5	37.7	47.3	50.5	53.9	48.1	51.0	**37.7**
10th QEF FMN	13.4	18.7	31.8	37.6	36.1	31.8	36.9	30.3	37.7	47.6	50.5	54.4	47.6	52.0	**37.6**
multicondition training	test set														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline	19.2	22.4	28.5	34.0	34.0	30.0	33.9	45.0	43.9	47.2	46.3	51.2	46.6	50.0	**38.0**
10th FMN	21.4	18.3	24.2	30.0	26.2	25.4	29.7	32.5	35.5	41.8	42.8	45.2	41.3	43.4	**32.7**
10th QE FMN	20.5	18.1	24.3	28.9	25.3	25.6	29.5	32.2	33.6	41.7	42.5	43.5	40.8	42.3	**32.1**
10th QEF FMN	19.9	17.7	24.1	29.0	25.0	25.6	28.9	29.8	33.0	39.8	42.2	43.8	40.4	42.6	**31.5**

Conclusions:

- The tests with the Aurora reference systems confirm that quantile equalization is generally applicable independent from the recognition task, the sampling rate, and the recognizer used.
- No specific optimizations of the recognizer used as back–end are required when applying quantile equalization.
- The results obtained are comparable to those reported by other groups using different approaches.

6.4 Quantile Equalization: Alternative Setups

6.4.1 Individual and Pooled Training Quantiles

When the estimation of the training quantiles was described on page 41 in section 5.3 it was noted that the recommended approach is to pool the training quantiles Q_i^{train} over all filter channels k. Even though the data's distribution on typical training sets (table 6.27) does obviously depend on the channel k, neglecting these differences only has a small influence on the error rates.

Table 6.27: Individual training quantiles for the different filter channels estimated on the clean training set of the Aurora 4 database.

	training quantiles				
filter channel	0	1	2	3	4
1	0.64	0.82	1.00	1.09	1.24
2	0.69	0.87	1.10	1.24	1.36
3	0.74	0.90	1.12	1.27	1.44
4	0.77	0.92	1.12	1.31	1.53
5	0.79	0.93	1.09	1.30	1.59
6	0.78	0.92	1.06	1.26	1.60
7	0.79	0.93	1.05	1.22	1.58
8	0.80	0.93	1.05	1.19	1.58
9	0.81	0.93	1.04	1.19	1.58
10	0.83	0.96	1.06	1.20	1.57
11	0.84	0.97	1.09	1.23	1.57
12	0.86	0.98	1.10	1.23	1.55
13	0.87	0.98	1.10	1.23	1.54
14	0.88	0.99	1.12	1.25	1.56
15	0.89	1.00	1.13	1.26	1.57
16	0.91	1.01	1.14	1.27	1.61
17	0.93	1.02	1.14	1.27	1.63
18	0.93	1.01	1.11	1.23	1.65
19	0.93	1.00	1.08	1.19	1.65
20	0.94	1.00	1.07	1.17	1.65
average	0.83	0.95	1.09	1.23	1.55

The results on the Car Navigation database presented in table 6.28 show that the use of individual training quantiles only leads to a negligible improvement of maximally 0.3% percentage points. On the Aurora 3 SpeechDat Car database (table 6.29) there even is a small deterioration of the results. This corresponds to the result of an other digit string recognition experiment in car environment (MoTiV database) presented in [Hilger and Ney 2001].

The situation could be expected to be different for a more complex large vocabulary speech recognition system with better acoustic modelling, but table 6.30 reveals that this

is not the case either. Even with more detailed acoustic models there is no significant difference between individual and pooled training quantiles. In the case of clean training the average error rate is only reduced from 25.9% to 25.7% and there is no improvement at all in the multicondition training case.

Conclusion:

- There is there is no need to use filter specific training quantiles. Even though the data's distribution usually differs in the individual filter channels, taking into account these differences does not significantly improve the final recognition results.

Table 6.28: Comparison of quantile equalization with pooled and individual training quantiles on the Car Navigation database. LOG: logarithm, CMN: cepstral mean normalization, 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization.

test set SNR [dB]			Word Error Rates [%]		
			office 21	city 9	highway 6
LOG		CMN	2.9	31.6	74.2
10th		FMN	2.8	19.9	40.1
10th	QE pooled	FMN	3.2	11.7	20.1
10th	QE individual	FMN	3.1	11.4	19.8

Table 6.29: Recognition results on the Aurora 3 SpeechDat Car database, the error rates shown for the different languages are weighted averages over the three conditions well matched, medium mismatch and high mismatch. LOG: logarithm, 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization.

			Word Error Rates [%]					rel. impr.
			Finnish	Spanish	German	Danish	average	[%]
LOG			24.6	20.8	16.9	31.7	**23.5**	0.0
10th		FMN	12.9	11.6	13.9	23.1	**15.4**	21.3
10th	QE pooled	FMN	11.1	10.8	12.9	19.8	**13.7**	28.9
10th	QE individual	FMN	11.2	10.9	12.9	20.1	**13.8**	28.5

Table 6.30: Average recognition results on the Aurora 4 noisy WSJ 5k database. LOG: logarithm, CMN: cepstral mean normalization, 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization.

			Word Error Rates [%]	
			clean training	multi. training
LOG		CMN	45.7	19.5
10th		FMN	29.7	17.8
10th	QE pooled	FMN	25.9	17.1
10th	QE individual	FMN	25.7	17.1

6.4.2 Different Numbers of Quantiles

The amount of data available for estimation determines how detailed and reliable the estimation of the cumulative distribution can be. Here the influence of number of quantiles N_Q on the final recognition result shall be investigated.

The number of quantiles determines how many addends are considered during the minimization of the transformation parameters θ_k (equation 5.8 on page 45). The lowest and the highest quantiles, i.e. the current minimal and maximal values of the signal, are not taken into account. These outliers should not have any influence on the parameter estimation. So if the total number of quantiles is N_Q the number of quantiles that are actually considered is $N_Q - 1$.

When $N_Q = 2$ only one quantile, namely the median, is used for the actual parameter estimation. Even with this setup mean and variance normalization are outperformed and the resulting error rates on the Car Navigation database (table 6.31) are already comparable to those of the recommended standard setup with $N_Q = 4$. The estimation of the transformation factors is just based on the median if $N_Q = 2$, but the transformation itself (equation 5.6) still is a non–linear power function: 0 and the maximal value are not transformed, and in between the each value is transformed differently, this can explain why quantile equalization just based on the median can still outperform the linear mean and variance normalization.

The amount of data the isolated utterances provide is small, so using more quantiles (6 or 12) has no positive influence on the results for the Car Navigation database: while there is no significant effect on the results if the transformation is restricted to individual filter channels (QE in table 6.31), the error rates even rise when the filter channels are combined (QEF).

On the Aurora 4 database the average length of the utterances is about 8 seconds, giving the utterance wise quantile equalization more data to reliably estimate the quantiles. Thus increasing the number of quantiles from 2 to 4 can reduces the error rate of the system trained on clean data by 0.5% percentage points from 26.4% to 25.9% (table 6.32). But even with 8 seconds of data using more than 4 quantiles does not have any significant positive effect.

Conclusions:

- Even when only calculating two quantiles and just considering the median during the calculation of the transformation parameters the resulting non–linear transformation can outperform linear mean and variance normalization.
- Using four quantiles $N_Q = 4$ can be recommended as standard setup, it can be used on short windows as well as complete utterances.
- Increasing the number of quantiles can not significantly improve the recognition results.

Table 6.31: Recognition results on the Car Navigation database for different numbers of quantiles. 10th: root instead of logarithm, FM(V)N: filter mean (and variance) normalization, QE N_Q: quantile equalization with N_Q quantiles, QEF quantile equalization with filter combination.

test set SNR [dB]	Word Error Rates [%]		
	office 21	city 9	highway 6
10th FMN	2.8	19.9	40.1
10th QE 2 FMN	3.3	11.7	19.9
10th QE 4 FMN	3.2	11.7	20.1
10th QE 6 FMN	3.2	11.9	20.4
10th QE 12 FMN	3.2	11.8	20.3
10th QEF 2 FMN	3.0	10.5	17.2
10th QEF 4 FMN	3.6	10.3	17.1
10th QEF 6 FMN	3.8	10.7	17.7
10th QEF 12 FMN	4.3	11.3	18.6

Table 6.32: Varying the number of quantiles. Average recognition results on the Aurora 4 noisy WSJ 5k database. 10th: root instead of logarithm, FMN: filter mean normalization, QE N_Q: quantile equalization with N_Q quantiles, QEF quantile equalization with filter combination.

	Word Error Rates [%]	
	clean training	multi. training
10th FMN	29.7	17.8
10th QE 2 FMN	26.4	17.2
10th QE 4 FMN	25.9	17.1
10th QE 6 FMN	25.8	17.1
10th QE 12 FMN	25.8	17.1
10th QEF 2 FMN	26.4	17.2
10th QEF 4 FMN	25.5	17.0
10th QEF 6 FMN	25.4	16.9
10th QEF 12 FMN	25.4	16.8

6.4.3 Application in Training

In the previous sections quantile equalization was only applied during the recognition tests. During the training only the usual mean normalization was applied. The reference quantiles were estimated on the training data without transforming it.

This approach can be a practical advantage in distributed speech recognition applications. The noise robustness of new terminals (e.g. mobile phones) can be enhanced by adding quantile equalization to the feature extraction front–end, without requiring an update of the other terminals and a retraining of the server side recognition system is not necessary either.

The results in the previous section have confirmed that this approach works in principle. Considerable error rate reductions can already be obtained by applying quantile equalization in recognition alone. Can these results be improved by also transforming the training data?

In experiments with full histogram normalization [Molau et al. 2001] it was shown that histogram normalization has to be applied to the training data of the system to get optimal recognition performance. From a theoretical point of view transforming the training data and the test data to the same canonical condition [Molau et al. 2001] should yield the best results. The corresponding approach when using quantile equalization is to estimate the training quantiles in one pass over the training data, before transforming it to match these average quantiles in a second pass.

The Car Navigation database confirms the assumption that the application in training can improve the best results (table 6.33). While the error rate on the clean data remains unchanged there are considerable improvements on the noisy test data sets: from 9.6% to 7.7% on the city data and from 17.1% to 15.1% on the highway data.

Unfortunately improvements in this order of magnitude could not be verified on other databases. On the Aurora 3 database the average word error rate could not be improved significantly (table 6.34) and the situation is similar on Aurora 4 (table 6.35). In the case of multicondition data, were the application in training could be expected to have the biggest impact, there is no influence on the error rates. When training on clean data the error rate does change, but while it is reduced for normal quantile equalization (QE) it rises from 25.1% to 26.4% when neighboring filter channels are combined (QEF).

Quantile equalization is only based on a rough estimate of the current data distribution, so applying it to clean training data can result in a distortion of the data and in the case of noisy training data some of the desired variability might get lost.

Conclusions:

- Applying quantile equalization in training does not lead to consistent improvements on all databases it was tested on.
- It is sufficient to apply quantile equalization in recognition only.

Table 6.33: Car Navigation database: quantile equalization applied in recognition only compared to the application in training too. 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF(2): quantile equalization with filter combination (2 neighbors).

test set SNR [dB]	Word Error Rates [%]		
	office 21	city 9	highway 6
10th FMN	2.8	19.9	40.1
10th QE FMN	3.2	11.7	20.1
10th QEF FMN	3.6	10.3	17.1
10th QEF2 FMN	3.6	9.6	17.1
10th QE train FMN	3.3	10.1	16.7
10th QEF train FMN	3.6	8.8	15.9
10th QEF2 train FMN	3.6	7.7	15.1

Table 6.34: Recognition results on the Aurora 3 SpeechDat Car database, the error rates shown for the different languages are averaged over the three conditions well matched, medium mismatch and high mismatch. 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization.

	Word Error Rates [%]				
	Finnish	Spanish	German	Danish	average
LOG	24.6	20.8	16.9	31.7	23.5
10th QE FMN	11.1	10.8	12.9	19.8	13.7
10th QE train FMN	10.7	10.7	12.8	22.3	13.6

Table 6.35: Quantile equalization in recognition and training. Average recognition results on the Aurora 4 noisy WSJ 5k database. 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization.

	Word Error Rates [%]	
	clean training	multi. training
10th FMN	29.7	17.8
10th QE FMN	25.9	17.1
10th QEF FMN	25.5	17.0
10th QE train FMN	25.1	17.0
10th QEF train FMN	26.4	17.0

6.4.4 Different Transformation Functions

The standard transformation function that was introduced in chapter 5 and applied so far was a power function combined with a linear term:

$$\tilde{Y_k} = T_k(Y_k, \theta_k) = Q_{kN_Q} \left(\alpha_k \left(\frac{Y_k}{Q_{kN_Q}} \right)^{\gamma_k} + (1 - \alpha_k) \frac{Y_k}{Q_{kN_Q}} \right) \tag{6.2}$$

In this section different alternative functions are investigated. First fixed values for α or γ will be studied. Keeping one value fixed reduces the search effort of the grid search.

When $\alpha_k = 1$ the transformation is simply a power function corresponding to a gamma–correction in image processing. If $\gamma_k = 2$ the computational complexity of the transformation is reduced significantly: the power function call can be replaced by a simple multiplication, allowing an implementation for embedded systems without floating point unit.

The tests on the Car Navigation database show that the three alternatives: variable alpha and gamma (QE), fixed alpha (QE $\alpha_k = 1$) and fixed gamma (QE $\gamma_k = 2$) perform equally well (table 6.36). The differences are negligible. On the Aurora 4 database the results are similar (table 6.38). Fixing the value $\alpha_k = 1$ or $\gamma_k = 2$ only has a minor effect on the error rates. The overall conclusion than can be drawn is that if the transformation is applied after the 10th root compression can be restricted by fixing α_k or γ_k if the computational requirements shall be reduced or an integer implementation is required.

If quantile equalization shall be added to an existing system with a baseline MFCC feature extraction that uses the logarithm the situation is different. Then the linear summand in equation 6.2 is crucial, because in that case quantile equalization has to be applied before the logarithm, to make sure that the incoming values are positive. Using $\gamma_k = 2$ is still not problematic but $\alpha_k = 1$ significantly increases the error rates (table 6.37), especially on the clean test set. If the linear summand in the transformation function is missing in the power function will scale small values further down towards zero — applying the logarithm afterwards will then distort the signal by enhancing the little amplitude differences of these small values.

From the computational requirements point of view quantile equalization in the filter–bank domain is favorable because the dimensionality of the feature vectors is already reduced considerably. If the reduction of the computational requirements is not the important issue, quantile equalization can also be applied directly on the magnitude spectrum. The expectation is that quantile equalization might be more effective if it is applied before the reduction of the dimensionality. But this expectation is not met (QE spect. in table 6.38), the transformation on the spectrum does not improve the results, even if frequency specific training quantiles are used.

Conclusions:

- If applied after the 10th root the transformation function (equation 6.2) can be simplified by holding one of the parameters constant without significantly affecting the recognition results.

- Only when applying quantile equalization before the logarithm the linear summand in equation 6.2 is important.
- Quantile equalization in the spectral domain only increases the required computations without improving the results.

Table 6.36: Car Navigation database: comparison of the standard transformation (equation 6.2) to restricted transformations with fixed transformation parameters. 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization.

test set SNR [dB]	Word Error Rates [%]		
	office 21	city 9	highway 6
10th FMN	2.8	19.9	40.1
10th QE FMN	3.2	11.7	20.1
10th QE $\alpha = 1$ FMN	3.2	11.8	20.2
10th QE $\gamma = 2$ FMN	3.2	12.0	20.1

Table 6.37: Car Navigation database: comparison of different transformation functions applied before the logarithm. 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization.

test set SNR [dB]	Word Error Rates [%]		
	office 21	city 9	highway 6
LOG FMN	3.5	27.0	63.9
QE LOG FMN	4.5	15.5	23.1
QE $\alpha = 1$ LOG FMN	11.0	18.3	23.5
QE $\gamma = 2$ LOG FMN	4.0	15.3	23.7

Table 6.38: Comparison of the standard transformation (equation 6.2) to restricted transformations. Average recognition results on the Aurora 4 noisy WSJ 5k database. 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, spect. pooled/individual: application in the spectral domain with pooled or frequency specific training quantiles.

	Word Error Rates [%]	
	clean training	multi. training
10th FMN	29.7	17.8
10th QE FMN	25.9	17.1
10th QE $\alpha = 1$ FMN	26.1	17.2
10th QE $\gamma = 2$ FMN	26.4	17.2
QE spect. pooled 10th FMN	26.1	17.1
QE spect. individual 10th FMN	25.9	17.2

6.4.5 Quantile Equalization with Different Root Functions

In section 6.2.2 the influence of different root functions that were used instead of the logarithm is investigated. There the conclusion was drawn that the 10th root is a good compromise that works well in many training and test conditions and it was used for all further tests. The situation might be different after the application of quantile equalization. The investigation in this section shall show if an optimization of the root functions can improve the best quantile equalization results obtained so far with the 10th root.

The 5th root combined with filter–bank mean normalization performed significantly better than the 10th root on the noisy test sets of the Car Navigation database (FMN table 6.39 and section 6.2.2). This supports the expectation that the combination of the 5th root with quantile equalization should also yield the best results, but the results show that this is not the case (QE FMN). After the application of quantile equalization the 5th root looses its advantage over the 10th root and the combination of neighboring (QEF FMN) filter channels leads to an increase of the error rate for the 5th root. The 10th root clearly yields the minimal error rates on the noisy data sets.

Table 6.39: Comparison of the logarithm in the feature extraction with different root functions on the Car Navigation database. 2nd – 20th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination.

test set SNR [dB]	Word Error Rates [%]		
	office 21	city 9	highway 6
20th FMN	**2.6**	22.0	49.2
10th FMN	2.8	19.9	40.1
5th FMN	3.3	**17.4**	**28.7**
2nd FMN	16.7	40.8	70.0
20th QE FMN	**2.9**	**11.7**	**19.9**
10th QE FMN	3.2	11.7	20.1
5th QE FMN	3.6	12.1	20.5
20th QEF FMN	**3.0**	11.0	18.1
10th QEF FMN	3.6	**10.3**	**17.1**
5th QEF FMN	5.1	12.9	19.1

A similar observation can be made on the Aurora 4 database when using the clean training data. As long as mean normalization is applied alone, using the 5th root instead of the 10th leads to an error rate reduction from 29.7% to 24.5% which is a relative improvement of 18% (table 6.40). After the application of quantile equalization the difference is not completely gone, but the relative improvement is reduced to 7%, the word error rate is only decreased from 25.5% to 23.7%. If multicondition training data is used quantile equalization does not change the observation that was already made in section 6.2.2: the different root functions (for except the 2nd root) yield similar results.

Conclusions:

- Quantile equalization reduces the differences between the root functions that were observed before, when applying mean normalization alone.
- The 10th root can be recommended as compromise that should be used in all situations.

Table 6.40: Comparison of the logarithm in the feature extraction with different root functions on the Aurora 4 database. 2nd – 20th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination.

			Word Error Rates [%]	
			clean training	multi. training
20th		FMN	33.2	18.0
10th		FMN	29.7	**17.8**
5th		FMN	**24.5**	18.0
2nd		FMN	30.7	26.3
20th	QEF	FMN	28.8	17.0
10th	QEF	FMN	25.5	**17.0**
5th	QEF	FMN	**23.7**	17.4

6.4.6 Utterance Wise, Two Pass, and Online Processing

The characteristics of the data that shall be recognized determine whether an utterance wise implementation of joint quantile equalization and mean normalization performs better than a moving window online implementation. If the SNR is constant over the utterance, taking into account more data for the estimation of the transformation parameters is likely to yield better results because the estimates will be more reliable. As soon as the SNR changes within an utterance the situation is different.

As already pointed out in the database description the Car Navigation database was collected in real driving conditions. The objective was to record realistic data without explicitly waiting for stationary conditions, so many recordings were made during acceleration, deceleration, gear shifts and changes of the road surface. Even though the isolated word utterances themselves are short there is an obvious change of the background noise level in many of them (figure 6.10). Under these circumstances the online implementation of mean normalization with 500ms delay and a short 1s window performs significantly better than the utterance wise version (10th FMN in table 6.42). Quantile equalization can reduce the difference between the utterance wise and the moving window implementations, but the online implementation still always yields better results (10th QEF2 FMN) on the noisy test data sets.

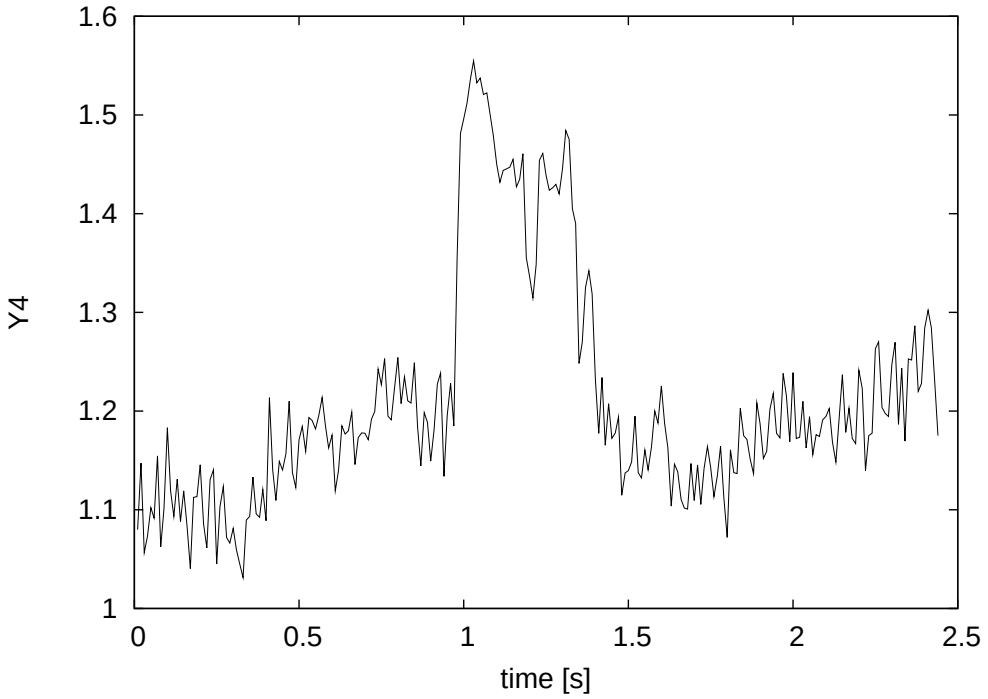

Figure 6.10: Output of the 4th Mel scaled filter after 10th root for an utterance from the Car Navigation test set. The level of the background noise changes during the recording.

If online processing is not required, a two pass approach can be used. The percentage of silence is determined in a first recognition pass, then the appropriate target quantiles can be calculated by combining the training quantiles estimated on the speech and silence portions of the signal respectively. In the case of full histogram normalization this ap-

proach significantly improves the recognition performance (table 6.19 on page 89). When using quantile equalization there is no consistent improvement (10th QE(F) FMN sil in table 6.42). The approximation of the cumulative distributions with four quantiles is rough. When determining the transformation function only the quantiles 1 to 3 are taken into account (table 6.41). They do not change in an extent that significantly influences the transformation function and consistently reduces the resulting error rates.

Table 6.41: Target quantiles for different amounts of silence (Car Navigation database)

	target quantiles				
silence [%]	0	1	2	3	4
0	0.99	1.09	1.19	1.31	1.52
25	0.99	1.08	1.16	1.26	1.47
50	0.98	1.07	1.13	1.22	1.42
75	0.98	1.05	1.10	1.17	1.37
100	0.98	1.04	1.07	1.13	1.32

The Aurora 4 database does not consist of recordings in realistic background noise conditions, it was created by adding noise to existing clean studio recordings. Some of the noises that were added are non–stationary, but the SNR remains constant over the utterances which leads to a different tendency in the results: The lowest error rates are obtained with utterance wise estimation of the transformation functions, online processing leads to an increase of the word error rates (Table 6.43). For quantile equalization with filter combination in the 1s delay 5s window setup the error rate rises from 25.9% to 27.3% for clean training and from 17.1% to 17.8% for multicondition training, but this is still better than the utterance wise baseline.

The 5s window means that for many utterances in the test set the processing is incremental, the end of the sentence is reached before the first frame is dropped at the end of the window. When using a 2 second window instead, the system should be able to react faster in changing noise conditions, but since the average SNRs are constant over the utterances in the Aurora 4 database the 2 second window does not perform better than the 5 second window.

Conclusions:

- Quantile equalization can be implemented using moving windows with a short delay, if the application requires real–time online processing.
- In real world conditions with changing SNR a moving window implementation that can adapt to these changes is recommendable, even if real–time response is not needed.
- A two pass approach that considers the amount of silence is not able improve quantile equalization.
- If the SNR of the utterances that are to be recognized can be expected to be constant and real–time possessing is not required, utterance wise processing yields the best results.

Table 6.42: Car Navigation database: utterance wise (UTTERANCE) quantile equalization compared to an online implementation (delay: window length). 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF(2): quantile equalization with filter combination (2 neighbors), QE sil: target quantiles dependent on the amount of silence.

		Word Error Rates [%]		
	test set SNR [dB]	office 21	city 9	highway 6
UTTERANCE	10th FMN	2.9	29.8	60.1
	10th QE FMN	3.0	12.0	19.4
	10th QEF FMN	3.7	11.1	17.5
	10th QEF2 FMN	3.4	11.3	18.2
UTTERANCE 2 PASS	10th QE sil FMN	3.0	11.8	19.7
	10th QEF sil FMN	3.4	10.5	18.1
0.5s : 1s	10th FMN	2.8	19.9	40.1
	10th QE FMN	3.2	11.7	20.1
	10th QEF FMN	3.6	10.3	17.1
	10th QEF2 FMN	3.6	9.6	17.1

Table 6.43: Comparison of utterance wise (UTTERANCE) and online implementations (delay: window length) of quantile equalization. Average recognition results on the Aurora 4 noisy WSJ 5k database. 10th: root instead of logarithm, FMN: filter mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination.

		Word Error Rates [%]	
		clean training	multi. training
UTTERANCE	10th FMN	29.7	17.8
	10th QE FMN	25.9	17.1
	10th QEF FMN	25.9	17.1
1s : 5s	10th FMN	31.0	18.3
	10th QE FMN	27.5	17.9
	10th QEF FMN	27.2	17.8
1s : 2s	10th FMN	31.1	18.4
	10th QE FMN	28.0	18.2
	10th QEF FMN	27.8	18.2

6.4.7 Combination of Quantile Equalization and MLLR

In this section the combination of quantile equalization and maximum likelihood linear regression (MLLR) [Leggetter and Woodland 1995] shall be investigated. Here a two pass offline setup is used to estimate the MLLR transformation matrices. In a first recognition pass over the data of each speaker session an initial alignment path is estimated. Based on this path the transformation matrices are calculated and then applied to transform the model parameters during the second recognition pass.

The actual number of transformation matrices that are applied depend on the number of observations in the first recognition pass. A transformation matrix is assigned to each leaf of a phonetic classification and regression tree that has a sufficient number of observations [Gales 1996]. For the experiments on the Aurora 4 database presented below this number is set to 2000, leading to an average of about 10 transformation matrices for a speaker session. Quantile equalization is applied in an utterance wise way (cf. section 6.3).

Table 6.44 shows the recognition results on the Aurora 4 noisy Wall Street Journal Database. In the case of clean training the two pass MLLR approach alone (10th FMN MLLR in table 6.44) already outperforms the best result previously obtained with quantile equalization. The average word error rate is 22.2% (unpublished internal results by Michael Pitz, Lehrstuhl für Informatik VI, RWTH Aachen, January 2004) as compared to 25.5% with quantile equalization and filter combination (10th QEF FMN).

Comparing the results for the individual noise conditions shows that the word error rate reduction through MLLR is especially large on the data sets with a microphone mismatch (conditions 8–14, cf. table 6.3 for the list of conditions). Test set 8 (clean, no additional noise, only a microphone mismatch) stands out. The word error rate with MLLR is 11.7% while the best result with quantile equalization is 21.2%.

Apparently MLLR can very well compensate the microphone distortions that are constant over the speakers session. On the other hand quantile equalization can compensate the noise that changes from an utterance to the next, so combining quantile equalization and MLLR yields a significant improvement over either of the individual methods alone. The final result with quantile equalization using filter combination and MLLR is 20.1% (10th QEF FMN MLLR), when training on clean data.

The situation is different for the setup with multicondition training data. Since there is no single training condition to which the MLLR matrix can transform the model parameters, there are no consistent improvements over quantile equalization through MLLR. The average word error rate with MLLR is not better than that obtained with quantile equalization and the combination of both methods does not improve the final results any further. The result with MLLR alone is 17.2% as compared to 17.0%, the best result with quantile equalization. Putting everything together the result with quantile equalization using filter combination and MLLR is 17.3% (10th QEF FMN MLLR).

A look at the individual noise conditions shows that MLLR leads to a significant deterioration of the recognition results in a few cases, e.g. noise condition 4 (10th FMN MLLR). These outliuers can be explained by singular transformation matrices. Even though the minimal number of observations for the estination of a transformation matrix

Table 6.44: Recognition results for the Aurora 4 noisy WSJ 16kHz databases. LOG: logarithm, CMN: cepstral mean normalization, 10th: 10th root instead of logarithm, FMN: filter–bank mean normalization, QE: quantile equalization, QEF: quantile equalization with filter combination, MLLR: maximum likelihood linear regression. Utterance wise mean normalization and quantile equalization. Speaker session wise two pass maximum likelihood linear regression.

clean training				Word Error Rates [%]												
				test set												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline				14.9	65.2	69.2	63.1	72.3	69.4	73.2	61.3	81.7	82.5	75.4	83.8	81.0	84.1	**69.8**
LOG		CMN		4.5	12.5	45.3	51.6	51.7	36.2	54.8	23.1	35.4	62.3	66.0	71.8	55.3	69.8	**45.7**
10th		FMN		4.4	8.6	20.8	27.5	28.7	20.9	30.1	22.3	28.9	41.9	45.3	48.6	41.4	46.3	**29.7**
10th	QE	FMN		4.3	8.3	16.2	21.1	22.1	19.7	22.8	22.0	27.8	37.2	40.7	41.1	39.0	40.5	**25.9**
10th	QEF	FMN		4.2	7.7	15.7	21.3	21.6	18.9	23.1	21.2	26.6	36.7	40.3	40.7	38.4	40.1	**25.5**
10th		FMN	MLLR	4.0	10.5	18.3	20.1	20.8	19.7	23.5	11.7	19.4	32.7	33.6	32.5	28.9	34.8	**22.2**
10th	QE	FMN	MLLR	3.9	9.2	16.1	19.7	18.3	18.9	21.5	12.3	17.0	29.2	31.9	29.3	27.9	30.8	**20.4**
10th	QEF	FMN	MLLR	3.9	9.1	15.6	19.5	17.2	18.6	21.2	10.3	17.1	29.5	31.0	27.9	28.1	32.1	**20.1**

multicondition training				test set												
				1	2	3	4	5	6	7	8	9	10	11	12	13	14	average
ISIP baseline				23.5	21.9	29.2	34.9	33.7	33.0	35.3	49.3	45.2	49.2	48.8	51.7	49.9	49.0	**39.6**
LOG		CMN		8.3	6.8	13.9	17.5	17.9	12.8	17.9	14.1	17.1	28.0	31.2	31.2	24.7	31.2	**19.5**
10th		FMN		7.2	6.9	12.4	15.1	16.3	12.2	15.4	14.4	16.7	24.8	27.5	28.3	23.1	28.3	**17.8**
10th	QE	FMN		7.1	7.1	11.3	14.7	14.6	11.6	14.1	14.3	16.9	24.2	27.5	27.0	23.1	25.6	**17.1**
10th	QEF	FMN		7.0	6.9	11.5	14.9	14.5	11.6	14.1	14.4	16.9	23.8	27.4	27.0	22.8	25.7	**17.0**
10th		FMN	MLLR	6.1	6.8	11.4	22.3	15.0	10.9	17.9	9.4	11.7	21.7	33.3	24.0	23.3	26.4	**17.2**
10th	QE	FMN	MLLR	6.3	6.7	34.3	13.5	16.3	10.6	13.0	9.4	11.7	24.8	23.5	32.3	21.2	20.8	**17.4**
10th	QEF	FMN	MLLR	6.0	6.5	35.4	13.6	16.2	10.6	12.9	9.0	11.6	24.9	24.6	24.0	23.2	23.8	**17.3**

was set to 2000, singular matrices occured. This shows that using a large number of obervations is not sufficient to rule out signular transformation matrices if the system was trained on multicondition data. An additional mechanism that checks the matrices for singularity and eventually falls back to a broader transformation class with a non–singular matirx would be required.

Conclusions:

- Clean training data:
  - The two pass speaker session wise MLLR approach outperforms quantile equalization.
  - MLLR yields the largest error rate reductions on test sets that have a constant microphone mismatch in addition to the varying noise.
  - Utterance wise quantile equalization and MLLR can be combined leading to results that are better than those that can be obtained with either method alone.
- Multicondition training data:
  - MLLR leads to a small average improvement over the baseline result.
  - Compared to quantile equalization MLLR does not lead to consistent improvements on all noise conditions.
  - The combination of quantile equalization and MLLR has no significant effect.
  - A fallback mechanism that is applied in the case of singular transformation matrices should be used to prevent outliers with high error rates.

6.5 Summary: Experimental Results

The general conclusions that can be drawn from the experimental evaluations are the following:

- Replacing the logarithm in the feature extraction by a root function x^r significantly increased the recognition performance on noisy data (section 6.2.2). The 10th root $r = 0.1$ was found to yield good results on all databases. For systems that shall recognize noisy data while being trained on clean data the 5th root $r = 0.2$ can be recommended.

- Quantile equalization, with a power function applied to individual filter bank–channels improved the recognition results on all noisy databases (section 6.3). The relative improvement depended on the baseline error rate and the amount of mismatch between the training and test conditions. In cases with a minor mismatch and a low baseline word error rate, the relative improvement was in the order of 5%. In high mismatch cases more than 50% relative improvement were observed.

- Concerning the setup of quantile equalization algorithm an important conclusion is that four quantiles $N_Q = 4$ can already be reliably estimated with little data, even in a moving window online setup. Increasing the number of quantiles if more data is available does not necessarily improve the results, so four quantiles are also sufficient to process longer sentences in an utterance wise setup (section 6.4.2).

- It was shown that the training quantile used as reference can be pooled, they do not have to be estimated individually for each filter channel (section 6.4.1).

- Applying quantile equalization to the training data only improved the recognition performance on one database (section 6.4.3). It can be applied during training, but it does not have to.

- Taking into account the dependencies between the filter channels, by combining the neighbors in a second transformation step was found to be helpful. Especially in the case of band limited noises, like car noise, significant improvements were observed (e.g. table 6.12 on page 81).

- For real–time online applications, the two quantile equalization steps were combined with the standard mean normalization in a way, that did not induce more delay than mean normalization itself has (section 5.4).

- The only parameter that had to be optimized empirically when changing to an other database was the overestimation factor with which the largest quantile was scaled (page 44). In all cases the optimal value was between 1.0 and 1.5.

The specific recognition results on the three databases that were investigated primarily can be summarized as follows:

- **Car Navigation Database:** On this isolated word database the effect of the acoustic model alone could be investigated. The mismatch between the training data recorded in a quiet office and the test data recorded in cars was particularly high, so the use of the 10th root and quantile equalization with filter combination had a particularly large effect. The application of quantile equalization in training was also able to improve the results. Compared to the baseline MFCC feature extraction with cepstral mean normalization, the word error rate was reduced from 31.6% to 7.7% on the city traffic data and from 74.2% to 15.2% on the highway data (pages 62 and 107).

- **Aurora 3 SpeechDat Car:** For the tests on this database quantile equalization was added to a given standard MFCC feature extraction that did not contain any normalization. The HTK reference recognizer setup was left unaltered. The relative improvement through quantile equalization largely depended on the mismatch of the individual subsets of the databases. The weighted overall average word error rate, calculated according to the official evaluation scheme, was reduced from 23.5% to 13.7% (page 96). In these tests the acoustic models of the reference system were not detailed enough to take an advantage from the slight transformation through the combination neighboring filter channels.

- **Aurora 4 noisy Wall Street Journal:** On this database experiments were carried out with the given reference system and the RWTH large vocabulary speech recognizer. The reference baseline of the system trained on clean 16kHz data was 60.8% word error rate. This number could be reduced to 37.6% by adding quantile equalization to the reference system. The final result with the RWTH recognizer was 27.5%. The corresponding numbers for the system trained on multicondition data were 38.0% that could be reduced to 31.5% by adding quantile equalization to the reference system and 17.8% when using the RWTH system. Compared to the results reported by other groups, these results are competitive (pages 93 and 98).

- **Additional tests:** Some additional tests on other databases (pages 86 to 88), like the EMBASSI data and different databases recorded by Lucent Bell Labs, underlined the genericity of the quantile equalization method.

Chapter 7

Scientific Contributions

In this work, a feature extraction method to increase the noise robustness of automatic speech recognition systems was presented. The approach was based on the idea of reducing an eventual mismatch between the recognition and training data by applying a parametric transformation function.

Logarithm and root functions:

- Prior to the investigations on quantile equalization, experiments were carried out to investigate the influence of the function that reduces the dynamic range of the signal during the feature extraction. Usually a logarithm is used. The investigations in this work have shown that it should be replaced by a root function, e.g. a 10th root, when recognizing noisy data. While the recognition performance on clean data was found to be similar, the root functions performed significantly better in noisy conditions. The correlation between a clean signal and noise corrupted signals was shown to be higher if root functions were applied.

Quantile Based Histogram Equalization:

- The quantile equalization algorithm which was presented is a modification of the non–parametric histogram equalization method. No specific assumptions about how the noise actually influences the speech have to be made, and a speech silence detection is not required. The cumulative distributions of the data were approximated with a small number of quantiles. It was shown that this approach has the advantage of being suited for real–time sigle pass online applications that only allow a short delay. It can also be used in offline applications in which the environment conditions change from one utterance to the next. The experimental results confirmed that the quantiles, which are independent from the scaling and range of the data, can already be reliably estimated from seconds of data.

  Based on these quantiles, the parameters of the actual transformations functions were estimated. Regarding the type of transformation function used to be used and the position in the feature extraction, the approach is general too. In this work a

two–step transformation, applied after the Mel–scaled filter–bank, was proposed. In the first step the individual filter–channels were transformed with a power function, before neighboring filter channels were combined linearly. These two transformation steps were integrated into the feature extraction and combined with the moving window mean normalization without increasing the delay.

Quantile equalization is an independent feature domain approach that does not require any feedback from the recognizer. It can be used in distributed speech recognition applications, with the feature extraction on independent mobile terminals and server side speech recognition.

The experimental tests with different recognition systems have shown that no specific optimizations of the recognizer itself are required in order to use quantile equalization. The system does not even have to be retrained, if quantile equalization is added to a given system.

In the tests with several different databases quantile equalization was shown to be applicable independent from the complexity of the recognition task. Quantile equalization can be used for simple digit string recognition applications as well as for larger vocabulary tasks, like noisy Wall Street Journal 5k. It does not rely on the availability of clean training data to estimate a clean speech model. Task specific sample noises to train a suitable noise model are not required either. Whether the training data is clean or noisy does not influence the procedure itself. However, the method aims at reducing an eventual mismatch between the training and test data, so the relative reduction of the word error rate depends on the baseline word error rate and the amount of mismatch. In cases with a considerable mismatch due to band limited stationary noise the largest improvements were observed.

The overall conclusion that can be drawn is: quantile based histogram equalization does not only increase the noise robustness, the method itself is robust too. When changing to a new recognition task or using a different recognition system as back–end, it will work reliably without requiring complicated parameter optimizations.

Chapter 8

Outlook

Within this work the focus was put on investigating the influence of quantile based histogram equalization individually. Future work could be dedicated to an in depth invesigation of the combination of quantile equalization with other feature or model domain methods to increase the noise robustness. If the combination with other methods is not redundant and leads to additional improvements of the results, the question arises how these different approaches can then be merged into one unified algorithm?

Concerning the application of the root function during feature extraction an important question that was not addressed so far is: can the optimal exponent of the root function be determined automatically, using the training data and eventually a development test set? So far, different roots were simply tested on several databases and the conclusion was drawn that the 10th root seems to be a good compromise that works well in all situations. A discriminative framework can be thought of, in which the exponent of the root function is chosen to maximize the class separability or alternatively minimize the recognition error rate on the training data.

The role of the target distribution, to which the training data could be transformed when applying full or quantile based histogram equalization, can also be studied. In this work, the overall average distribution of the training data was used as target distribution for the individual utterances. As alternative the use of a Gaussian with zero mean and fixed variance has been suggested by other groups. With respect to minimizing the recognition error rate there could be more appropriate target distributions. There might even be a way of determining an optimal discriminative target distribution in a data driven framework.

Going an other step further the feature extraction procedure itself could be revised. The various different feature extraction steps, which require numerous parameters, could eventually be replaced by a more appropriate non–linear transformation function with fewer parameters. The parameters of that function should then be optimized in a way that makes distribution of the data in the actual feature space especially well suited for recognition.

Appendix A

Database Statistics

Table A.1: Database statistics: Car Navigation

Car Navigation				
Language	German			
Task	isolated words			
Speaking style	read			
Sampling frequency	16kHz			
Subsets	Training	Test		
Recording environment	office	office	car city	car highway
Total amount [h:min]	18:48	1:42	1:42	1:48
Silence percentage [%]	60	69	73	75
Number of speakers	86	14	14	14
Number of words	61742	2069	2100	2100
Recognizer vocabulary	–	2100	2100	2100

Table A.2: Database statistics: Aurora 2 – noisy TI digit strings

Aurora 2 – noisy TI digit strings		
Language	US English	
Task	digit strings	
Speaking style	read	
Sampling frequency	8kHz	
Recording environment	studio + added noises	
Subsets	2 × training	70 × test
Total amount [h:min]	5:06	0:35
Silence percentage [%]	26	24
Number of speakers	110	104
Number of sentences	8840	1001
Number of digits	27727	3257
Recognizer vocabulary	–	11

Table A.3: Database statistics: Aurora 3 – Danish SpeechDat Car

Aurora 3 – Danish SpeechDat Car						
Language	Danish					
Task	digit strings					
Speaking style	read					
Sampling frequency	8kHz					
Recording environment	car					
Subsets	well matched		medium mismatch		high mismatch	
	training	test	training	test	training	test
Total amount [h:min]	3:36	1:15	1:20	0:11	1:48	0:33
Silence percentage [%]	42	21	42	17	42	20
Number of speakers	416	396	300	103	416	328
Number of sentences	3440	1474	1245	204	1720	648
Number of digits	12468	5614	4593	805	6234	2516
Recognizer vocabulary	–	11	–	11	–	11

Table A.4: Database statistics: Aurora 3 – German SpeechDat Car

Aurora 3 – German SpeechDat Car						
Language	German					
Task	digit strings					
Speaking style	read					
Sampling frequency	8kHz					
Recording environment	car					
Subsets	well matched		medium mismatch		high mismatch	
	training	test	training	test	training	test
Total amount [h:min]	2:21	0:53	1:10	0:14	1:10	0:23
Silence percentage [%]	31	21	32	21	31	22
Number of speakers	144	64	137	33	144	56
Number of sentences	2032	897	997	241	1007	394
Number of digits	11448	5009	5574	1366	5662	2162
Recognizer vocabulary	–	11	–	11	–	11

Table A.5: Database statistics: Aurora 3 – Finnish SpeechDat Car

Aurora 3 – Finnish SpeechDat Car						
Language	Finnish					
Task	digit strings					
Speaking style	read					
Sampling frequency	8kHz					
Recording environment	car					
Subsets	well matched		medium mismatch		high mismatch	
	training	test	training	test	training	test
Total amount [h:min]	7:43	2:07	2:24	0:25	3:51	0:48
Silence percentage [%]	59	35	58	36	59	36
Number of speakers	237	231	147	88	237	174
Number of sentences	3080	1320	963	248	1540	496
Number of digits	17778	7698	5606	1462	8889	2830
Recognizer vocabulary	–	11	–	11	–	11

Table A.6: Database statistics: Aurora 3 – Spanish SpeechDat Car

Aurora 3 – Spanish SpeechDat Car						
Language	Spanish					
Task	digit strings					
Speaking style	read					
Sampling frequency	8kHz					
Recording environment	car					
Subsets	well matched		medium mismatch		high mismatch	
	training	test	training	test	training	test
Total amount [h:min]	5:12	2:02	2:25	1:10	2:36	0:51
Silence percentage [%]	38	32	38	32	39	32
Number of speakers	229	102	217	114	229	85
Number of sentences	3392	1522	1607	850	1696	631
Number of digits	18334	8056	8652	4543	9167	3325
Recognizer vocabulary	–	11	–	11	–	11

Table A.7: Database statistics: Aurora 4 – noisy Wall Street Journal 5k

Aurora 4 – noisy WSJ 5k		
Language	US English	
Task	newspaper articles	
Speaking style	read	
Sampling frequency	16kHz	
Recording environment	studio + added noises	
Subsets	2 × training	14 × test
Total amount [h:min]	15:08	0:21
Silence percentage [%]	19	26
Number of speakers	83	8
Number of sentences	7138	166
Number of words	129435	2737
Recognizer vocabulary	–	5061
Trigram perplexity	–	62.3

Table A.8: Database statistics: EMBASSI

EMBASSI			
Language	German		
Task	command phrases in natural language		
Speaking style	read, with hesitations, false starts and interruptions		
Sampling frequency	16kHz		
Recording environment	quiet and with background noise		
Subsets	training segmented	quiet segmented	4 × noisy unsegmented
Total amount [min]	57	14	~ 7
Silence percentage [%]	25	22	~ 40
Number of speakers	12	6	6
Number of recordings	1440	360	6
Number of words	8306	2088	~ 920
Recognizer vocabulary	—	474	
Trigram perplexity	—	145	

Appendix B

Mathematical Symbols and Acronyms

B.1 Mathematical Symbols

In following all symbols that are used throughout the text are listed. Some additional symbols that only appear once, e.g. in the introduction, are explained in the context in which they are used.

w_1^N	sequence of words $w_1^N = w_1, \ldots, w_N$
N	number of words
x_1^T	sequence of feature vectors $x_1^T = w_1, \ldots, w_T$
t, T	time frame index, number of time frames
$p(w_1^N \vert x_1^T)$	probability of a word sequence given the acoustic feature vectors
$p(x_1^T \vert w_1^N)$	acoustic model probability of the feature vectors given a word sequence
$p(w_1^N)$	language model probability, prior probability of the word sequence
s, l	HMM–state (mixture) index, density index
c_{sl}	density weight
μ_{sl}	mean vector of density sl
Σ_{sl}	covariance matrix
$\mathcal{N}(x \vert \mu, \Sigma)$	Gaussian Normal distribution
PP	perplexity

Y	filter–bank output vector (after logarithm or root)
Y_k	kth component of the filter–bank output vector
k	filter channel index
$P(\cdot)$	cumulative distribution function
$P^{train}(\cdot)$	cumulative distribution of the training data
Q	arbitrary quantile
Q_{ki}	ith quantile of filter channel k during recognition
Q_{ki}^{train}	ith training quantile of filter channel k
$T(Y, \theta)$	power function transformation
$T_k(Y_k, \theta_k)$	power function transformation of filter channel k
θ	transformation parameters $\theta = \{\alpha, \gamma\}$
α	balancing factor
γ	exponent of the power function
$\tilde{Y}$	filter–bank output vector power function transformation
$\tilde{T}(\tilde{Y}, \tilde{\theta})$	combination of neighboring filter channels
$\tilde{\theta}$	transformation parameters $\tilde{\theta} = \{\lambda, \rho\}$
λ	combination factor for the left neighbor
ρ	combination factor for the right neighbor
cor	correlation coefficient

B.2 Acronyms

10th 10th root $x^{0.1}$

20th 20th root $x^{0.05}$

2nd 2nd root, square root $x^{0.5}$

5th 5th root $x^{0.2}$

AMD Advanced Micro Devices (company)

ARPA Advanced Research Projects Agency

CDCN Codeword–Dependent Cepstral Normalization

CDF Cumulative Distribution Function

CMN Cepstral Mean Normalization

DARPA Defense Advanced Research Projects Agency

DEL DELetions

EMBASSI Elektronische Multimediale Bedien- und service ASSIstenz

ETSI European Telecommunications Standards Institute

FFT Fast Fourier Transform

FMN Filter–bank Mean Normalization

FMVN Filter–bank Mean and Variance Normalization

GSM Global System for Mobile communications

HM High Mismatch, training and test data partitioning of the Aurora 3 databases

HMM Hidden Markov Model

HN Histogram Normalization

HTK Hidden Markov model Toolkit [Woodland et al. 1995]

ICSI International Computer Science Institute, Berkeley, CA

INS INSertions

ISIP Institute for Signal and Information Processing, Mississippi State University

LDA Linear Discriminant Analysis

LOG LOGarithm

MFCC Mel–Frequency Cepstral Coefficients

MLLR Maximum Likelihood Linear Regression

MM Medium Mismatch, data partitioning of the Aurora 3 databases

MP3 Moving Picture experts group layer 3, audio file format

OGI Oregon Graduate Institute for Science and Engineering, Portland, OR

PDA Personal Digital Assistant

PP language model PerPlexity

QE Quantile Equalization

QEF Quantile Equalization with neighboring filter combination

QEF2 Quantile Equalization with filter combination, 2 left and right neighbors

RASTA RelAtive SpecTrAl filtering

ROT feature space ROTation

RWTH Rheinisch–Westfälische Technische Hochschule, Aachen, Germany

SIMD Single Instruction Multiple Data

SNR Signal–to–Noise Ratio

SUB SUBstitutions

TI Texas Instruments (company)

VUI Voice User Interface

WER Word Error Rate

WI007 Work Item 007 in the Aurora evaluations (baseline MFCC front–end)

WM Well Matched, data partitioning of the Aurora 3 databases

WSJ Wall Street Journal

Bibliography

[Acero 1993] A. Acero, *Acoustical And Environmental Robustness In Automatic Speech Recognition*, Kluwer Academic Publishers Group, Boston, MA, 1993.

[Adami et al. 2002] A. Adami, L. Burget, S. Dupont, H. Garudadri, F. Grezl, H. Hermansky, P. Jain, S. Kajarekar, N. Morgan, and S. Sivadas, "Qualcomm–ICSI–OGI features for ASR," in *Proc. of the International Conference on Spoken Language Processing 2002*, Denver, CO, September 2002, vol. 1, pp. 21–24.

[Afify and Siohan 2001] M. Afify and O. Siohan, "Sequential noise estimation with optimal forgetting for robust speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2001*, Salt Lake City, UT, May 2001, vol. I, pp. 229–232.

[Afify et al. 2001] M. Afify, H. Jiang, F. Korkmazskiy, C.-H. Lee, Q. Li, O. Siohan, F. K. Soong, and A. C. Surendran, "Evaluating the Aurora connected digit recognition task – a Bell Labs approach," in *Proc. of the European Conference on Speech Communication and Technology 2001*, Aalborg, Denmark, September 2001, vol. 1, pp. 633–637.

[Aikawa et al. 1993] K. Aikawa, H. Singer, H. Kawahara, and Y. Tohkura, "A dynamic cepstrum incorporating time–frequency masking and ita application to continuous speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1993*, Minneapolis, MN, April 1993, vol. II, pp. 668–671.

[Andrassy et al. 2001] B. Andrassy, F. Hilger, and C. Beaugeant, "Investigations on the combination of four algorithms to increase the noise robustness of a DSR front-end for real world car data," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding 2001*, Madonna di Campiglio, Trento, Italy, December 2001, p. 4.

[Atal 1974] B. S. Atal, "Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification," *Journal of the Acoustical Society of America*, vol. 55, pp. 1304–1312, 1974.

[Aubert and Ney 1995] X. Aubert and H. Ney, "Large vocabulary continuous speech recognition using word graphs," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1995*, Detroit, MI, USA, May 1995, vol. 1, pp. 49–52.

[Bahl et al. 1983] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, pp. 179–190, March 1983.

[Baker 1975] J. K. Baker, "Stochastic modeling for automatic speech understanding," in *Speech Recognition*, D. R. Reddy, Ed., pp. 512–542. Academic Press, New York, NY, USA, 1975.

[Bakis 1976] R. Bakis, "Continuous speech recognition via centisecond acoustic states," in *Proc. of the 91st Meeting Acoustical Society of America 1976*, Washington, DC, April 1976.

[Balchandran and Mammone 1998] R. Balchandran and R. J. Mammone, "Non-parametric estimation and correction on non-linear distortion in speech systems," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1998*, Seattle, WA, September 1998, vol. II, pp. 749–752.

[Ballard and Brown 1982] D. H. Ballard and C. M. Brown, *Computer Vision*, pp. 70–71, Prentice-Hall, Englewood Cliffs, NJ, 1982.

[Baum and Petrie 1966] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," in *Annals of Mathematical Statistics 1966*, 1966, vol. 37, pp. 1554–1563.

[Bayes 1763] T. Bayes, "An essay towards solving a problem in the doctrine of chances," *Philosopical Transactions of the Royal Society of London*, vol. 53, pp. 370–418, 1763, Reprinted in *Biometrika*, vol. 45, no. 3/4, pp. 293–315, December 1958.

[Bellman 1957] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.

[Benitez et al. 2001] C. Benitez, L. Burget, B. Chen, S. Dupont, H. Garudadri, H. Hermansky, P. Jain, S. Kajarekar, and S. Sivadas., "Robust ASR front-end using spectral-based and discriminant features: experiments on the Aurora tasks," in *Proc. of the European Conference on Speech Communication and Technology 2001*, Aalborg, Denmark, September 2001, vol. 1, pp. 429–432.

[Bernstein and Shallom 1991] A. Bernstein and I. Shallom, "An hypothesized Wiener filtering approach to noisy speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1991*, Toronto, Canada, May 1991, vol. I, pp. 913–916.

[Berouti et al. 1979] M. Berouti, R. Schwarz, and J. Makhoul, "Enhancement of speech corrupted by noise," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1979*, Washington, DC, April 1979, vol. I, pp. 208–211.

[Beulen et al. 1997] K. Beulen, E. Bransch, and H. Ney, "State-tying for context dependent phoneme models," in *Proc. of the European Conference on Speech Communication and Technology 1997*, Rhodes, Greece, September 1997, vol. 3, pp. 1447–1450.

[Bocchieri 1993] E. Bocchieri, "Vector quantization for the efficient computation of continuous density likelihoods," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1993*, Minneapolis, MN, USA, April 1993, vol. 2, pp. 692–695.

[Bogert et al. 1963] B. P. Bogert, M. J. R. Healy, and J. W. Tuckey, "The quefrency alanysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking," in *Proc. of the Symposium Time Series Analysis 1963*, M. John Wiley Rosenblatt and Sons, Eds., New York, NY, 1963, pp. 209–243.

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

[Brown et al. 1992] P. Brown, V. Della Pietra, P. de Souza, J. Lai, and R. Mercer, "Class-based n-gram models of natural language," *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[Cerisara et al. 2000] C. Cerisara, L. Rigazio, R. Bomann, and J.-C. Junqua, "Transformation of Jacobian matricies for noisy speech recognition," in *Proc. of the International Conference on Spoken Language Processing 2000*, Beijing, China, October 2000, vol. I, pp. 369–372.

[Cooke et al. 1996] M. Cooke, A. Morris, and P. Green, "Recognising occluded speech," in *Proc. of the ESCA Workshop on the Auditory Basis of Speech Perception 1996*, Keele, UK, July 1996, pp. 297–300.

[Cooley and Tuckey 1965] J. W. Cooley and J. W. Tuckey, "An algorithm for the machine calculation of complex Fourier series," *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, April 1965.

[Davis and Mermelstein 1980] S. B. Davis and P. Mermelstein, "Comparison of parametric representations of monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357–366, August 1980.

[Dempster et al. 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[Deshmukh et al. 1999] N. Deshmukh, A. Ganapathiraju, J. Hamaker, J. Picone, and M. Ordowski, "A public domain speech–to–text system," in *Proc. of the European Conference on Speech Communication and Technology 1999*, Budapest, Hungary, September 1999, vol. 5, pp. 2127–2130.

[Dharanipragada and Padmanabhan 2000] S. Dharanipragada and M. Padmanabhan, "A nonlinear unsupervised adaptation technique for speech recognition," in *Proc. of the International Conference on Spoken Language Processing 2000*, Beijing, China, October 2000, vol. IV, pp. 556–559.

[Duda and Hart 1973] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, NY, USA, 1973.

[ETSI 2000] "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; front–end feature extraction algorithms; compression algorithms," Tech. Rep., ETSI ES 201 108 V1.1.2, Sophia Antipolis, France, April 2000.

[ETSI 2002] "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced feature extraction algorithm; compression algorithms," Tech. Rep., ETSI ES 202 050 V0.1.1, Sophia Antipolis, France, April 2002.

[Le Floc'h et al. 1992] A. Le Floc'h, R. Salami, B. Mouy, and J. Adoul, "Evaluation of linear and non–linear spectral subtraction methods for enhancing noisy speech," in *Proc. of the ESCA Workshop ETRW on Speech Processing in Adverse Conditions 1992*, Cannes, France, November 1992, pp. 131–134.

[Fukunaga 1990] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Academic Press, Boston, 2nd edition, 1990.

[Furui 1981] S. Furui, "Cepstral analysis technique for automatic speaker verification," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 254–272, April 1981.

[Gales and Young 1996] M. J. F. Gales and S. J. Young, "Robust continuous speech recognition using parallel model combination," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 5, pp. 352–359, September 1996.

[Gales 1995] M. J. F. Gales, *Model-Based Techniques For Noise Robust Speech Recognition*, Ph.D. thesis, University of Cambridge, Cambridge, UK, September 1995.

[Gales 1996] M. Gales, "The generation and use of regression class trees for MLLR adaptation," Tech. Rep. CUED/F-INFENG/TR263, Cambridge University, 1996, Available via anonymous ftp from:svr-ftp.eng.cam.ac.uk.

[Gales 1998] M. J. F. Gales, "Predictive model–based compensation schemes for robust speech recognition," *Speech Communication*, vol. 1–3, no. 25, pp. 49–75, August 1998.

[Generet et al. 1995] M. Generet, H. Ney, and F. Wessel, "Extensions to absolute discounting for language modeling," in *Proc. of the European Conference on Speech Communication and Technology 1995*, Madrid, Spain, September 1995, vol. 2, pp. 1245–1248.

[Gnanadesikan 1980] R. Gnanadesikan, *Handbook of Statistics: Analysis of Variance*, vol. 1, chapter 5, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, 1980.

[Haderlein and Nöth 2003] T. Haderlein and E. Nöth, "The EMBASSI speech corpus," Tech. Rep., Chair for Pattern Recognition (Informatik 5), University of Erlangen–Nürnberg, December 2003.

[Haderlein et al. 2003] T. Haderlein, G. Stemmer, and E. Nöth, "Speech recognition with μ-law companded features on reverberated signals," in *Proc. of Text, Speech and Dialogue: 6th International Conference, Ceské Budejovice, Czech Republic, September*

2003, V. Matoušek and P. Mautner, Eds., vol. 2807 of *Lecture Notes in Computer Science*, pp. 173–180. Springer–Verlag, Berlin, November 2003.

[Haeb-Umbach et al. 1998] R. Haeb-Umbach, X. Aubert, P. Beyerlein, D. Klakow, M. Ullrich, A. Wendemuth, and P. Wilcox, "Acoustic modeling in the Philips Hub-4 continuous-speech recognition system," in *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop 1998*, Lansdowne, VA, February 1998, p. 4.

[Hastie et al. 2001] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, chapter 3.4.3, Springer Verlag, New York, 2001.

[Hermansky and Morgan 1994] H. Hermansky and N. Morgan, "Rasta processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, October 1994.

[Hermansky et al. 1991] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "Compensation of the effect of the communication channel in auditory like analysis of speech," in *Proc. of the European Conference on Speech Communication and Technology 1991*, Genova, Italy, September 1991, vol. 3, pp. 1367–1370.

[Hilger and Ney 2000] F. Hilger and H. Ney, "Noise level normalization and reference adaptation for robust speech recognition," in *Proc. of the International Workshop on Automatic Speech Recognition: Challenges for the new Millenium 2000*, Paris, France, September 2000, pp. 64–68.

[Hilger and Ney 2001] F. Hilger and H. Ney, "Quantile based histogram equalization for noise robust speech recognition," in *Proc. of the European Conference on Speech Communication and Technology 2001*, Aalborg, Denmark, September 2001, vol. 2, pp. 1135–1138.

[Hilger and Ney 2003] F. Hilger and H. Ney, "Evaluation of quantile based histogram equalization with filter combination on the Aurora 3 and 4 databases," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 341–344.

[Hilger et al. 2002] F. Hilger, S. Molau, and H. Ney, "Quantile based histogram equalization for online applications," in *Proc. of the International Conference on Spoken Language Processing 2002*, Denver, CO, September 2002, vol. 1, pp. 237–240.

[Hilger et al. 2003] F. Hilger, H. Ney, O. Siohan, and F. K. Soong, "Combining neighboring filter channels to improve quantile based histogram equalization," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2003*, Hong Kong, China, April 2003, vol. I, pp. 640–643.

[Hirsch and Pearce 2000] H.-G. Hirsch and D. Pearce, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. of the International Workshop on Automatic Speech Recognition: Challenges for the new Millenium 2000*, Paris, France, September 2000, pp. 181–188.

[Hirsch et al. 1991] H.-G. Hirsch, P. Meyer, and H. Ruehl, "Improved speech recognition using high–pass filtering of subband envelopes," September 1991.

[Hirsch 1993] H.-G. Hirsch, "Estimation of noise spectrum and its application to SNR-estimation and speech enhancement," Tech. Rep. TR-93-012, Berkeley, CA, 1993.

[Hirsch 2002] H.-G. Hirsch, "Experimental framework for the performance evaluation of speech recognition front–ends on a large vocabulary task, Version 2.0, AU/417/02," Tech. Rep., ETSI STQ-Aurora DSR Working Group, October 2002.

[Hon and Lee 1991] H. W. Hon and K. F. Lee, "Recent progress in robust vocabulary-independent speech recognition," in *Proceedings DARPA Speech and Natural Language Processing Workshop 1991*, Pacific Grovem, USA, 1991, pp. 258–263.

[Huang and Jack 1989] X. D. Huang and M. A. Jack, "Semi-continuous hidden Markov models for speech signals," *Computer, Speech, and Language*, vol. 3, no. 3, pp. 329–252, 1989.

[Huang et al. 1990] X. D. Huang, K. F. Lee, and H. W. Hon, "On semi-continuous hidden Markov modeling," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1990*, Albuquerque, NM, USA, April 1990, pp. 689–692.

[Huo et al. 1997] Q. Huo, H. Jiang, and C.-H. Lee, "A Bayesian predictive classification approach to robust speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1997*, Munich, Germany, April 1997, vol. II, pp. 1547–1550.

[Hwang et al. 1993] M. Y. Hwang, X. D. Huang, and F. Alleva, "Predicting unseen triphones with senones," Tech. Rep. 510.7808 C28R 93-139 2, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1993.

[Jardino 1996] M. Jardino, "Multi-lingual stochastic n-gram class language models," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1996*, Atlanta, GA, USA, May 1996, vol. 1, pp. 161–163.

[Jelinek 1969] F. Jelinek, "A fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, vol. 13, pp. 675–685, November 1969.

[Jelinek 1976] F. Jelinek, "Continuous speech recognition by statistical methods," *Proc. of the IEEE*, vol. 64, no. 10, pp. 532–556, April 1976.

[Jelinek 1991] F. Jelinek, "Self-organized language modeling for speech recognition," in *Readings in Speech Recognition*, A. Waibel and K. F. Lee, Eds., pp. 450–506. Morgan Kaufmann Publishers, San Mateo, CA, USA, 1991.

[Jelinek 1997] F. Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambrige, MA, 1997.

[Jiang et al. 1998] H. Jiang, K. Hirose, and Q. Huo, "A minimax search algorithm for CDHMM based robust speech recognition," in *Proc. of the International Conference on Spoken Language Processing 1998*, Sydney, Australia, November 1998, vol. 2, pp. 389–392.

[Junqua et al. 1995] J.-C. Junqua, D. Fohr anf J.-F. Mari, T. Appelbaum, and B. Hanson, "Time derivatives, cepstral normalization and spectral parameter filtering for continuously spelled names over the telephone," in *Proc. of the European Conference on Speech Communication and Technology 1995*, Madrid, Spain, September 1995, pp. 1385–1388.

[Junqua et al. 1999] J.-C. Junqua, S. C. Fincke, and K. L. Field, "The Lombard effect: A reflex to better communicate with others in noise," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1999*, Phoenix, AZ, September 1999, vol. IV, pp. 2083–2086.

[Junqua 1993] J.-C. Junqua, "The Lombard reflex and its role on human listeners and automatic speech recognizers," *Journal of the Acoustical Society of America*, vol. 93, no. 1, pp. 510–524, 1993.

[Kanthak et al. 2000a] S. Kanthak, K. Schütz, and H. Ney, "Using SIMD instructions for fast likelihood calculation in LVCSR," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2000*, Istanbul, Turkey, June 2000, vol. III, pp. 1531–1534.

[Kanthak et al. 2000b] S. Kanthak, A. Sixtus, S. Molau, R. Schlüter, and H. Ney, "Fast search for large vocabulary speech recognition," in *Verbmobil: Foundations of Speech-to-Speech Translation*, W. Wahlster, Ed., pp. 63–78. Springer Verlag, Berlin, Germany, 2000.

[Katz 1987] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *IEEE Transactions on Speech and Audio Processing*, vol. 35, pp. 400–401, March 1987.

[Keysers et al. 2000a] D. Keysers, J. Dahmen, and H. Ney, "A probabilistic view on tangent distance," in *22. DAGM Symposium Mustererkennung 2000*, Kiel, Germany, September 2000, pp. 107–114, Springer.

[Keysers et al. 2000b] D. Keysers, J. Dahmen, T. Theiner, and H. Ney, "Experiments with an extended tangent distance," in *Proceedings 15th International Conference on Pattern Recognition 2000*, PBarcelona, Spain, September 2000, vol. 2, pp. 38–42.

[Kim et al. 2003] Y. J. Kim, H. W. Kim, W. Lim, and N. S. Kim, "Feature compensation technique for robust speech recognition in noisy environments," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 357–360.

[Kim 1998] N. S. Kim, "Nonstationary environment compensation based on sequential estimation," *IEEE Signal Processing Letters*, vol. 5, no. 3, pp. 57–59, March 1998.

[Klakow 1998] D. Klakow, "Language-model optimization by mapping of corpora," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1998*, Seattle, WA, USA, May 1998, vol. 2, pp. 701–704.

[Kneser and Ney 1993] R. Kneser and H. Ney, "Improved clustering techniques for class-based statistical language modeling," in *Proc. of the European Conference on Speech Communication and Technology 1993*, Berlin, Germany, 1993, vol. 2, pp. 973–976.

[Korkmazskiy et al. 2000] F. Korkmazskiy, F. K. Soong, and O. Siohan, "Constrained spectrum normalization for robust speech recognition in noise," in *Proc. of the International Workshop on Automatic Speech Recognition: Challenges for the new Millenium 2000*, Paris, France, September 2000, pp. 64–68.

[Kuhn and De Mori 1990] R. Kuhn and R. De Mori, "A cache-based natural language model for speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 570–583, June 1990.

[Lee and Huo 1999] C.-H. Lee and Q. Huo, "Adaptive classification ond decision strategies for robust speech recognition," in *Proc. of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions 1999*, Tampere, Finland, May 1999, pp. 45–52.

[Lee 1998] C.-H. Lee, "On stochastic feature and model compensation approaches to robust speech recognition," *Speech Communication*, vol. 25, pp. 29–47, August 1998.

[Leggetter and Woodland 1995] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech and Language*, vol. 9, no. 4, pp. 806–814, 1995.

[Levinson et al. 1983] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell System Technical Journal*, vol. 62, no. 4, pp. 1035–1074, April 1983.

[Li et al. 2001] Q. Li, F. K. Soong, and O. Siohan, "An auditory system-based feature for robust speech recognition," in *Proc. of the European Conference on Speech Communication and Technology 2001*, Aalborg, Denmark, September 2001, vol. 1, pp. 619–621.

[Lim and Oppenheim 1978] J. Lim and A. Oppenheim, "All pole modelling of degraded speech," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, no. 3, pp. 197–210, June 1978.

[Lindberg 2001] B. Lindberg, "Danish SpeechDat–Car database for ETSI STQ Aurora advanced DSR AU/378/01," Tech. Rep., Aalborg University, STQ Aurora DSR Working Group, January 2001.

[Liporace 1982] L. Liporace, "Maximum likelihood estimation for multi-variate observations of Markov sources," *IEEE Transactions on Information Theory*, vol. 28, no. 5, pp. 729–734, 1982 1982.

[Lockwood and Boudy 1992] P. Lockwood and J. Boudy, "Experiments with a nonlinear spectral subtractor (NSS), hidden Markov models and the projection for robust speech recognition in cars," *Speech Communication*, vol. 11, pp. 215–228, 1992.

[Macherey and Ney 2003] K. Macherey and H. Ney, "Features for tree-based dialogue course management," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 601–604.

[Macherey et al. 2001] W. Macherey, D. Keysers, J. Dahmen, and H. Ney, "Improving automatic speech recognition using tangent distance," in *Proc. of the European Conference on Speech Communication and Technology 2001*, Aalborg, Denmark, September 2001, vol. 3, pp. 1825–1828.

[Macho et al. 2002] D. Macho, L. Mauuary, B. Noe, Y. M. Cheng, D. Ealey, D. Jouvet, H. Kelleher, D. Pearce, and F. Saadoun, "Evaluation of a noise-robust DSR front-end on Aurora databases," in *Proc. of the International Conference on Spoken Language Processing 2002*, Denver, CO, September 2002, vol. 1, pp. 17–20.

[Macho 2000] D. Macho, "Spanish SDC–Aurora database for ETSI STQ Aurora WI008 advanced front–end evaluation: Description and baseline results AU/271/00," Tech. Rep., Universitat Politècnica de Catalunya, STQ Aurora DSR Working Group, November 2000.

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

[Martin et al. 1997] S. C. Martin, J. Liermann, and H. Ney, "Adaptive topic-dependent language modeling using word-based varigrams," in *Proc. of the European Conference on Speech Communication and Technology 1997*, Rhodes, Greece, September 1997, vol. 3, pp. 1447–1450.

[Martin et al. 1998] S. C. Martin, J. Liermann, and H. Ney, "Algorithms for bigram and trigram word clustering," *Speech Communication*, vol. 24, no. 1, pp. 19–37, 1998.

[Martin et al. 1999] S. C. Martin, C. Hamacher, J. Liermann, F. Wessel, and H. Ney, "Assessment of smoothing methods and complex stochastic language modeling," in *Proc. of the European Conference on Speech Communication and Technology 1999*, Budapest, Hungary, September 1999, vol. 4, pp. 1939–1942.

[Merhav and Lee 1993] N. Merhav and C.-H. Lee, "A minimax classification procedure for a universal adaptation method based on HMM–composition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 1, no. 1, pp. 90–100, January 1993.

[Molau et al. 2001] S. Molau, M. Pitz, and H. Ney, "Histogram based normalization in the acoustic feature space," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding 2001*, Madonna di Campiglio, Trento, Italy, December 2001, p. 4.

[Molau et al. 2002] S. Molau, F. Hilger, D. Keysers, and H. Ney, "Enhanced histogram normalization in the acoustic feature space," in *Proc. of the International Conference on Spoken Language Processing 2002*, Denver, CO, September 2002, vol. 1, pp. 1421–1424.

[Molau et al. 2003] S. Molau, F. Hilger, and H. Ney, "Feature space normalization in adverse acoustic conditions," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2003*, Hong Kong, China, April 2003, vol. I, pp. 656–659.

[Molau 2003] S. Molau, *Nomalization in the Acoustic Feature Space for Improved Speech Recognition*, Ph.D. thesis, RWTH Aachen – University of Technology, Aachen, Germany, February 2003.

[Moreno et al. 1996] P. J. Moreno, B. Raj, and R. M. Stern, "A vector Taylor series approach for environment-independent speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1996*, Atlanta, GA, May 1996, vol. II, pp. 733–737.

[Moreno 1996] P. J. Moreno, *Speech Recognition in Noisy Environments*, Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, April 1996.

[Morris et al. 1998] A. Morris, M. Cooke, and P. Green, "Some sollutions to the missing feature problem in data classification, with applications to noise robust ASR," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1998*, Seattle, WA, May 1998, vol. II, pp. 737–740.

[Nene and Nayar 1996] S. A. Nene and S. K. Nayar, "Closest point search in high dimensions," in *IEEE Conference on Computer Vision and Pattern Recognition 1996*, San Francisco, CA, USA, June 1996, pp. 859–865.

[Netsch 2001] L. Netsch, "Description and baseline results for the subset of the SpeechDat–Car German database used for ETSI STQ Aurora WI008 advanced front–end evaluation AU/273/00," Tech. Rep., Texas Instruments, STQ Aurora DSR Working Group, January 2001.

[Ney and Noll 1988] H. Ney and A. Noll, "Phoneme modeling using continuous mixture densities," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1988*, New York, NY, April 1988, pp. 437–440.

[Ney et al. 1987] H. Ney, D. Mergel, A. Noll, and A.Paeseler, "A data-driven organization of the dynamic programming beam search for continuous speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1987*, Dallas, TX, April 1987, vol. I, pp. 833–836.

[Ney et al. 1992] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder, "Improvements in beam search for 10000-word continuous speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1992*, San Francisco, CA, March 1992, vol. I, pp. 9–12.

[Ney et al. 1994] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependencies in language modeling," *Computer, Speech, and Language*, vol. 2, no. 8, pp. 1–38, 1994.

[Ney et al. 1997] H. Ney, S. C. Martin, and F. Wessel, "Statistical language modeling using leaving-one-out," in *Corpus Based Methods in Language and Speech Processing*, S. Young and G. Bloothooft, Eds. Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997.

[Ney et al. 1998] H. Ney, L. Welling, S. Ortmanns, K. Beulen, and F. Wessel, "The RWTH large vocabulary continuous speech recognition system," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1998*, Seattle, WA, May 1998, vol. II, pp. 853–856.

[Ney 1984] H. Ney, "The use of a one-stage dynamic programming algorithm for connected word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 2, pp. 263–271, April 1984.

[Ney 1990] H. Ney, "Acoustic modeling of phoneme units for continuous speech recognition," in *Signal Processing V: Theories and Applications*, L. Torres, E. Masgrau, and M. A. Lagunas, Eds., pp. 65–72. Elsevier Science Publishers, Amsterdam, The Netherlands, 1990.

[Ney 1993] H. Ney, "Search strategies for large-vocabulary continuous-speech recognition," in *Speech Recognition and Coding - New Advances and Trends, NATO Advanced Studies Institute, Bubion, Spain, June/July 1993*, A. J. Rubio Ayuso and J. M. Lopez Soler, Eds., pp. 210–225. Springer, Berlin, Germany, 1993.

[Nokia 2000] "Baseline results for subset of SpeechDat-Car Finnish database for ETSI STQ WI008 advanced front-end evaluation AU/225/00," Tech. Rep., Nokia, STQ Aurora DSR Working Group, January 2000.

[Nolazco-Flores and Young 1994] J. Nolazco-Flores and S. Young, "Continuous speech recognition in noise using spectral subtraction and HMM adaptation," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1994*, Adelaide, Australia, April 1994, vol. I, pp. 409–412.

[Obuchi and Stern 2003] Y. Obuchi and R. M. Stern, "Normalization of time–derivative parameters using histogram normalization," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 665–666.

[Odell et al. 1994] J. J. Odell, V. Valtchev, P. C. Woodland, and S. J. Young, "A one-pass decoder design for large vocabulary recognition," in *Proceedings ARPA Spoken Language Technology Workshop 1994*, Plainsboro, NJ, USA, March 1994, pp. 405–410.

[Openshaw and Mason 1994] J. Openshaw and J. Mason, "On the limitations of cepstral features in noise," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1994*, Adelaide, Australia, April 1994, vol. II, pp. 49–52.

[Ortmanns and Ney 1995] S. Ortmanns and H. Ney, "An experimental study of the search space for 20000-word speech recognition," in *Proc. of the European Conference on Speech Communication and Technology 1995*, Madrid, Spain, September 1995, vol. 2, pp. 901–904.

[Ortmanns et al. 1996] S. Ortmanns, H. Ney, and A. Eiden, "Language-model look-ahead for large vocabulary speech recognition," in *Proc. of the International Conference on Spoken Language Processing 1996*, Philadelphia, PA, October 1996, vol. 4, pp. 2095–2098.

[Ortmanns et al. 1997a] S. Ortmanns, H. Ney, and X. Aubert, "A word graph algorithm for large vocabulary continuous speech recognition," *Computer, Speech, and Language*, vol. 11, no. 1, pp. 43–72, January 1997.

[Ortmanns et al. 1997b] S. Ortmanns, H. Ney, and T. Firzlaff, "Fast likelihood computation methods for continuous mixture densities in large vocabulary speech recognition," in *Proc. of the European Conference on Speech Communication and Technology 1997*, Rhodes, Greece, September 1997, vol. 1, pp. 139–142.

[Ortmanns et al. 1997c] S. Ortmanns, L. Welling, K. Beulen, F. Wessel, and H. Ney, "Architecture and search organization for large vocabulary continuous speech recognition," in *Informatik 97: Informatik als Innovationsmotor*, M. Jarke, K. Pasedach, and K. Pohl, Eds., pp. 456–465. Springer, Berlin, Germany, 1997.

[Parihar and Picone 2002] N. Parihar and J. Picone, "DSR front end LVCSR evaluation AU/384/02," Tech. Rep., ETSI Aurora Working Group, Sophia Antipolis, France, December 2002.

[Parihar and Picone 2003] N. Parihar and J. Picone, "An analysis of the Aurora large vocabulary evaluations," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 337–340.

[Paul 1991] D. B. Paul, "Algorithms for an optimal A^* search and linearizing the search in the stack decoder," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1991*, Toronto, Canada, May 1991, vol. 1, pp. 693–696.

[Pearce 2000] D. Pearce, "Enabling new speech driven services for mobile devices: An overview of the ETSI standards activities for distributed speech recognition front-ends," in *Applied Voice Input/Output Society Conference 2000*, San Jose, CA, May 2000.

[Pellom and Hacioglu 2003] B. Pellom and K. Hacioglu, "Recent improvements in the CU SONIC ASR system for noisy speech: The SPINE task," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2003*, Hong Kong, China, April 2003, vol. I, pp. 4–7.

[Picone 1993] J. Picone, "Signal modeling techniques in speech recognition," *Proceedings of the IEEE*, vol. 81, no. 9, pp. 1215–1248, September 1993.

[Poynton 1993] C. A. Poynton, "Gamma and its disguises: The nonlinear mappings of intensity in perception, CRTs, film and video," *Society of Motion Picture and Television Engineers Journal*, vol. 102, no. 12, pp. 1099–1108, December 1993.

[Rabiner and Junag 1993] L. R. Rabiner and B. H. Junag, *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ, April 1993.

[Rabiner and Schafer 1978] L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice-Hall, Englewood Cliffs, NJ, 1978.

[Rabiner 1989] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. of the IEEE*, vol. 77, no. 2, pp. 257–286, February 1989.

[Raj et al. 1997] B. Raj, E. B. Gouvea, and R. M. Stern, "Vector polynomial approximations for robust speech recognition," in *Proc. of the ESCA Workshop ETRW on Speech Processing in Adverse Conditions 1997*, Pont-a-Mousson, France, April 1997, pp. 131–134.

[Rigazio et al. 2003] L. Rigazio, P. Nguyen, D. Kryze, and J.-C. Junqua, "Large vocabulary noise robustness on Aurora4," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 345–348.

[Rosenberg et al. 1994] A. E. Rosenberg, C.-H. Lee, and F. K. Soong, "Cepstral channel normalization techniques for HMM–based speaker verification," in *Proc. of the International Conference on Spoken Language Processing 1994*, Yokohama, Japan, September 1994, vol. I, pp. 1835–1838.

[Rosenfeld 1994] R. Rosenfeld, *Adaptive Statistical Language Modeling: A Maximum Entropy Approach*, Ph.D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, 1994.

[Sagayama et al. 1997] S. Sagayama, Y. Yamaguchi, and S. Takahashi, "Jacobian adaptation of noisy speech models," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding 1997*, Santa Barbara, CA, December 1997, pp. 396–403.

[Sagayama et al. 2001] S. Sagayama, K. Shinoda, M. Nakai, and H. Shimodaira, "Analytic methods for acoustic model adaptation: A review," in *Proc. of the Workshop on Adaptation Methods 2001*, Sophia Antipolis, France, August 2001, pp. 67–76.

[Sagayama 1999] S. Sagayama, "Differential approach to model adaptation," in *Proc. of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions 1999*, Tampere, Finland, May 1999, pp. 61–66.

[Sakoe 1979] H. Sakoe, "Two-level DP-matching - a dynamic programming-based pattern matching algorithm for connected word recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 27, pp. 588–595, December 1979.

[Schwartz and Austin 1991] R. Schwartz and S. Austin, "A comparison of several approximate algorithms for finding multiple (N-best) sentence hypotheses," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1991*, Toronto, Canada, May 1991, vol. 1, pp. 701–704.

[Schwartz and Chow 1990] R. Schwartz and Y.-L. Chow, "The N-best algorithm: An efficient and exact procedure for finding the N most likely sentence hypotheses," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1990*, Albuquerque, NM, USA, April 1990, pp. 81–84.

[Segura et al. 2002] J. C. Segura, M. C. Benitez, A. de la Torre, and A. J. Rubio, "Feature extraction combining spectral noise reduction and cepstral histogram equalization for robust ASR," in *Proc. of the International Conference on Spoken Language Processing 2002*, Denver, CO, September 2002, vol. 1, pp. 225–228.

[Segura et al. 2003] J. Segura, J. Ramirez, C. Bentiez, A. de la Torre, and A. J. Rubio, "Improved feature extraction based on spectral noise reduction and nonlinear feature normalization," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 353–356.

[Simard et al. 1992] P. Simard, Y. Le Cun, J. Denker, and B. Victorri, "An efficient algorithm for learning invariances in adaptive classifiers," in *Proceedings 11th International Conference on Pattern Recognition 1992*, The Hague, The Netherlands, August 1992, pp. 651–655.

[Singer et al. 1995] H. Singer, K. Paliwal, T. Beppu, and Y. Sagisaka, "Effect of RASTA–type processing for speech recognition with speaking rate mismatch," in *Proc. of the European Conference on Speech Communication and Technology 1995*, Madrid, Spain, September 1995, pp. 487–490.

[Siohan et al. 1999] O. Siohan, C. Chesta, and C.-H. Lee, "Hidden Markov model adaptation using maximum a posteriori linear regression," in *Proc. of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions 1999*, Tampere, Finland, May 1999, pp. 87–90.

[Siohan et al. 2000] O. Siohan, T.-A. Myrvoll, and C.-H. Lee, "Structural maximum a posteriori linear regression for fast HMM adaptation," in *Proc. of the International Workshop on Automatic Speech Recognition: Challenges for the new Millenium 2000*, Paris, France, September 2000, pp. 120–127.

[Sixtus et al. 2000] A. Sixtus, S. Molau, S. Kanthak, R. Schlüter, and H. Ney, "Recent improvements of the RWTH large vocabulary speech recognition system on spontaneous speech," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2000*, Istanbul, Turkey, June 2000, vol. III, pp. 1671–1674.

[Sixtus 2003] A. Sixtus, *Across-Word Phoneme Models for Large Vocabulary Continuous Speech Recognition*, Ph.D. thesis, RWTH Aachen – University of Technology, Aachen, Germany, January 2003.

[Stahl et al. 2000] V. Stahl, A. Fischer, and R. Bippus, "Quantile based noise estimation for spectral subtraction and wiener filtering," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2000*, Istanbul, Turkey, June 2000, vol. III, pp. 1875–1878.

[Stevens and Volkmann 1940] S. S. Stevens and J. Volkmann, "The relation of pitch to frequency: A revised scale," *American Journal of Psychology*, vol. 53, pp. 329–353, July 1940.

[Stevens et al. 1937] S. S. Stevens, J. Volkmann, and E. B. Newman, "A scale for the measurement of the psychological magnitude pitch," *Journal of the Acoustical Society of America*, vol. 8, pp. 185–190, January 1937.

[Stouten et al. 2003] V. Stouten, H. van Hamme, J. Duchateau, and P. Wambacq, "Evaluation of model–based feature enhancement on the Aurora-4 task," in *Proc. of the European Conference on Speech Communication and Technology 2003*, Geneva, Switzerland, September 2003, vol. 1, pp. 349–352.

[Surendran et al. 1996] A. C. Surendran, C.-H. Lee, and M. Rahim, "A maximum likelihood stochastic matching approach to non–linear equalization for robust speech recognition," in *Proc. of the International Conference on Spoken Language Processing 1996*, Philadelphia, PA, April 1996, vol. 3, pp. 1832–1835.

[Tian and Viikki 1999] J. Tian and O. Viikki, "Generalized cepstral analysis for speech recognition in noise," in *Proc. of the Workshop on Robust Methods for Speech Recognition in Adverse Conditions 1999*, Tampere, Finland, May 1999, pp. 87–90.

[de la Torre et al. 2002] A. de la Torre, J. C. Segura, M. C. Benitez, A. M. Peinado, and A. Rubio, "Non-linear transformations of the feature space for robust speech recognition," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2002*, Orlando, FL, May 2002, vol. I, pp. 401–404.

[van Kampen 1992] N. G. van Kampen, *Stochastic Processes in Physics and Chemistry*, Elsevier Science Publishers B. C., Amsterdam, The Netherlands, 1992.

[de Veth and Boves 1996] J. de Veth and L. Boves, "Comparison of channel normalization techniques for speech recognition over the phone," in *Proc. of the International Conference on Spoken Language Processing 1996*, Philadelphia, PA, October 1996, vol. 4, pp. 2332–2335.

[de Veth et al. 1998] J. de Veth, B. Cranen, and L. Boves, "Acoustic backing off in the local distance computation for robust automatic speech recognition," in *Proc. of the International Conference on Spoken Language Processing 1998*, Sydney, Australia, November 1998, vol. 4, pp. 1427–1430.

[de Veth et al. 2001] J. de Veth, B. Cranen, and L. Boves, "Acoustic backing-off as an implementation of missing feature theory," *Speech Communication*, vol. 34, no. 3, pp. 247–265, June 2001.

[Viikki and Laurila 1998] O. Viikki and K. Laurila, "Cepstral domain segmental feature vector normalization for noise robust speech recognition," *Speech Communication*, vol. 25, pp. 133–147, August 1998.

[Vintsyuk 1971] T. K. Vintsyuk, "Element-wise recognition of continuous speech composed of words from a specified dictionary," *Kibernetika*, vol. 7, pp. 133–143, March 1971.

[Viterbi 1967] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimal decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, 1967.

[Vizinho et al. 1999] A. Vizinho, P. Green, M. Cooke, and L. Josifovski, "Missing data theory, spectral subtraction and signal-to-noise estimation for robust ASR: An integrated study," in *Proc. of the European Conference on Speech Communication and Technology 1999*, Budapest, Hungary, September 1999, vol. 5, pp. 2407–2410.

[van Vuuren and Hermansky 1997] S. van Vuuren and H. Hermansky, "Data-driven design of rasta-like filters," in *Proc. of the European Conference on Speech Communication and Technology 1997*, Rhodes, Greece, September 1997, vol. 1, pp. 409–412.

[Weiss et al. 1974] M. Weiss, E. Aschkenasy, and T. Parsons, "Processing speech signals to attenuate interference," in *Proc. of the IEEE Symposion on Speech Recognition 1974*, Pittsburgh, PA, 1974, vol. April, pp. 292–293.

[Welling et al. 1997] L. Welling, N. Haberland, and H. Ney, "Acoustic front-end optimization for large vocabulary speech recognition," in *Proc. of the European Conference on Speech Communication and Technology 1997*, Rhodes, Greece, September 1997, vol. 4, pp. 2099–2102.

[Wessel et al. 1997] F. Wessel, S. Ortmanns, and H. Ney, "Implementation of word based statistical language models," in *Proceedings Spoken Queries in European Languages (SQEL) Workshop on Multi-Lingual Information Retrieval Dialogs 1997*, Pilsen, Czech Republic, April 1997, pp. 55–59.

[de Wet et al. 2003] F. de Wet, J. de Veth, B. Cranen, and L. Boves, "The impact of spectral and energy mismatch on the Aurora2 digit recognition task," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 2003*, Hong Kong, China, April 2003, vol. II, pp. 105–108.

[Wiener 1949] N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series with Engineering Applications*, MIT Press, Cambridge, MA, 1949.

[Wood 1996] J. Wood, "Invariant pattern recognition: A review," *Pattern Recognition*, vol. 29, no. 1, pp. 1–17, January 1996.

[Woodland et al. 1995] P. C. Woodland, C. J. Legetter, J. J. Odell, V. Valtchev, and S. J. Young, "The 1994 HTK large vocabulary speech recognition system," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1995*, Detroit, MI, USA, May 1995, vol. 1, pp. 573–576.

[Young et al. 1994] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modeling," in *Proceedings ARPA Human Language Technology Workshop 1994*, Plainsboro, NJ, USA, March 1994, pp. 307–312, Morgan Kaufmann Publishers.

[Young et al. 2000] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Vatchev, and P. Woodland, *The HTK Book (for HTK version 3.0)*, Cambridge University Engineering Department, Cambridge, UK, July 2000.

[Young 1992] S. J. Young, "The general use of tying in phoneme based HMM recognizers," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing 1992*, San Francisco, CA, USA, March 1992, vol. 1, pp. 569–572.

Lebenslauf

Florian Erich Hilger

Geboren am 7. August 1973
in Bonn – Bad Godesberg

Ausbildung

1979 – 1980	Yonkers Elementary School, Bronxville, NY, USA
1980 – 1981	Gemeinschaftsgrundschule Heiderhof, Bonn – Bad Godesberg
1981 – 1983	Löwenburgschule, Bad Honnef
1983 – 1985	Siebengebirgsgymnasium, Bad Honnef
1985 – 1989	Deutsche Schule Genf, Schweiz
1989 – 1992	Deutsche Schule Paris, Frankreich
Juni 1992	Abitur
1992 – 1998	Studium der Physik an der RWTH Aachen
	Nebenfächer im Grundstudium: Numerik, Informatik
	Nebenfächer im Hauptstudium: Statistische Physik, Biomedizinische Technik, Klinische Physiologie
	Diplomarbeit am Helmholtz–Institut für Biomedizinische Technik: „Modellierung der Bewegung oberer Extremitäten bei gegebener Muskelaktivierung“
Mai 1998	Diplom
1998 – 2004	Wissenschaftlicher Mitarbeiter im Bereich automatische Spracherkennung am Lehrstuhl für Informatik VI der RWTH Aachen

Praktische Tätigkeiten

1996 – 1998	Studentische Hilfskraft am I. Physikalischen Institut der RWTH Aachen Betreuung des Physikalischen Praktikums für Mediziner
1998 – 2004	Wissenschaftlicher Mitarbeiter am Lehrstuhl für Informatik VI, RWTH Aachen
Juni – Sept. 2002	Gastaufenthalt beim Multimedia Communications Research Lab, Bell Laboratories, Lucent Technologies, Murray Hill, NJ, USA
seit Feb. 2004	Projektmanager bei der Telenet GmbH Kommunikationssysteme in München