Rheinisch Westfälische Technische Hochschule Aachen
Lehrstuhl für Informatik VI
Prof. Dr.-Ing. Hermann Ney

Seminar
Speech Recognition and Language Processing im SS 2003

Histogram Normalization

Haihua Luo

Matrikelnummer 239706

July 30, 2003

Supervisor: Florian Hilger

Abstract

Acoustic variations usually cause a mismatch between training and recognition conditions and lead to a severe deterioration of the recognition performance of speech recognition systems. Histogram normalization can be applied during feature extraction to reduce this mismatch significantly. It transforms the signals to match the overall distribution in training and test data by using a histogram based approach. Quantile equalization is a parametric type of histogram normalization for online applications. It approximates the distributions with a few quantiles and a parametric transformation is applied to reduce the eventual mismatch. Experiments have shown that histogram normalization and quantile equalization are effective methods to increase the robustness of speech recognition systems under different acoustic conditions. Combined with other normalization techniques, further recognition performance improvements can be obtained.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Motivation

The paper is organized as follows. First automatic speech recognition and feature extraction are introduced in Chapter 1. In the next chapter variety of normalization technologies in feature space are explained, such as cepstral mean normalization and histogram normalization. Chapter 3 discusses histogram normalization method in detail. The basic algorithm is described and and the integration at different stages of the feature extraction is investigated. Algorithm of quantile based histogram equalization method is described and recognition experiments over various databases are presented in Chapter 4, followed by a conclusion in Chapter 5.

1.1 Automatic Speech Recognition

Speech is one of the most natural and effective tools to communicate among human beings, and also a straightforward method to express human's complicated thoughts. During the age of information, people desire to generate, transfer, store and achieve speech information more efficiently by machine. Amongst all kinds of research topics in this field Automatic Speech Recognition (ASR) is central to the future of human-machine interaction.

The speech recognition process is a wide range of research areas invoking knowledge of signal processing, physics (acoustics), pattern recognition, communication and information theory, linguistics, physiology, statistics, artificial intelligence, etc. Investigations in ASR by machine have been done for almost four decades. Although it is still far from achieving the desired goal of a machine that can understand all spoken languages by all speakers under different situations, scientists have developed all kinds of methods to reduce the error percentage during speech recognition. Histogram Normalization (HN) is one of those developing mechanisms performed in signal analysis.

From a viewpoint of Bayes decision rule the automatic speech recognition process consists of three steps in general (Figure 1.1) [14][16]:

1. Acoustic modelling

The acoustic model includes acoustic analysis, phoneme inventory and pronunciation lexicon. The acoustic analysis is also known as feature extraction or signal processing front-end. It parameterizes the speech input with a sequence of acoustic feature vectors $x_1 \cdots x_t$, so that further processing can recognize different acoustic signals according to their specific features. There is a large variety of the parametric representations, e.g., the short-time spectral envelope, the short time energy, zero crossing rates, Mel freqency cep-

stral coefficients (MFCC), and so on. The phoneme inventory and pronunciation lexicon provides the probability $P_r(x_1 \cdots x_t \mid w_1 \cdots w_n)$ of the observed acoustic signal given a hypothesized word sequence $w_1 \cdots w_n$.

2. Language modelling

The language model captures the linguistic properties of the language and provides the probability of a word sequence $P_r(w_1 \cdots w_n)$.

3. Global search

In this process the most probable sentence for the input speech that shall be recognized will be searched. Search implements Bayes decision rule on the basis of the acoustic model and the language model. To obtain the best recognition result, it needs to maximize the multiplication of $P_r(w_1 \cdots w_n)$ and $P_r(x_1 \cdots x_t \mid w_1 \cdots w_n)$.

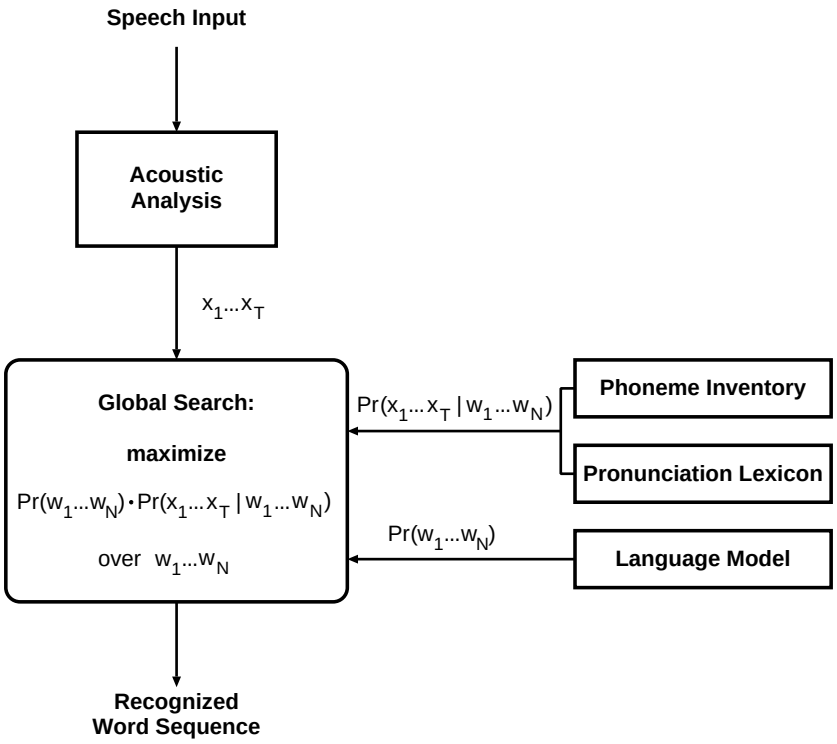

Figure 1.1: *Bayes decision rule in speech recognition*

1.2 Performance Criteria

The performance of an ASR system is usually evaluated by recognition error rate, robustness, and additional criteria, such as real time factor, memory requirements and software complexity.

1. Word error rate

Error rate is the most important and straightforward measure of speech recognition performance. Word error rate (WER) is commonly used to evaluate the performance of various techniques. It is defined by:

$$WER(\%) = \frac{Sub + Del + Ins}{Total} \times 100\%$$

where Sub, Del, Ins and $Total$ are the number of substitution, deletion, insertion errors and total number of recognition words, respectively.

2. Robustness

Robustness is also of great importance to a modern ASR system. The ultimate aim of robustness is to preserve high performance in speech recognition, regardless of variations faced in real life. The robustness of a recognition system is heavily influenced by the ability to handle the presence of background noise and to cope with the distortion by the frequency characteristic of the transmission channel.

1.3 MFCC Feature Extraction

The aim of feature extraction is to extract salient features from the speech waveform, which are critical for the acoustic modeling and recognition of the unknown sounds. Amongst various feature extraction techniques, the most popular one in current ASR systems is known as features of Mel-frequency cepstrum coefficients (MFCCs) [14]. A typical MFCC feature extraction process contains following stages, as illustrated in Figure 1.2 [14]:

1. Pre-emphasis

A sequence of raw speech signals is imported into feature extraction. A digital filter emphasizes high frequency components in the signals to imitate the unequal sensitivity of human hearing at different frequencies.

2. Segmenting and Windowing

The (pre-emphasized) speech signal is segmented into frames and a Hamming window is often employed to minimize the adverse effect of framing. In typical case every 10ms-segment a 25ms-wide Hamming window is applied to the speech samples.

3. Magnitude Spectrum

The short-time spectral analysis is performed via discrete Fourier transformation (DFT) or fast Fourier transformation (FFT) to convert speech frames to their frequency domain representation.

4. Mel Frequency Warping

The frequency contents in magnitude spectrum of the speech f_k are warped correspondently to Mel-scaled frequency $\tilde{f}_k$ with a sampling frequency:

$$\tilde{f}_k = 2595 \cdot \log_{10}\left(1 + \frac{f_k}{700Hz}\right)$$

5. Critical Band Integration

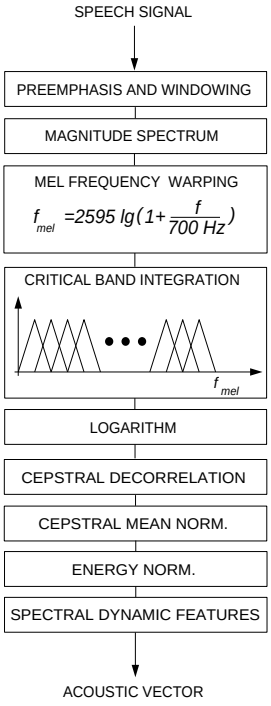

Figure 1.2: *Typical workflow of feature extraction*

The task in this step is to sum magnitude of Fourier components in each critical band with a triangular window.

6. Logarithm
Logarithm of the filter bank outputs is computed to compress the spectral envelope.

7. Cepstral Decorrelation
Discrete cosine transformation is performed on the logarithm of the magnitude of the filter bank output to get cepstral coefficients.

8. Cepstral Mean Normalization
This step subtracts mean to eliminate the influence of unknown transfer functions.

9. Calculation of the derivatives
The dynamic characteristics of MFCC features (delta cepstrum features) which is beneficial to ASR performance are calculated for further use in recognition.

Chapter 2

Feature Space Normalization Techniques

2.1 ASR in Adverse Environments

Despite the technological success achieved in a variety of applications, current ASR technology has not yet reached full maturity. The performance of speech recognition deteriorates significantly in real applications with variations of acoustic signals and environments. These undesired variables consist of noisy environments, distortion of transmission channels and speaker variability (speaking styles, accents), which result in a mismatch between training data and test utterance. In a word, large variability in the acoustic signal is the major error source for automatic speech recognition systems [11][13]. For the sake of the robustness of speech recognition system, it is necessary to minimize the mismatch.

The techniques proposed to cope with the mismatch can be roughly divided into two broad categories, normalization schemes and adaptation techniques. The normalization schemes try to reduce the mismatch by transforming the acoustic vectors, while the adaptation techniques attempt to adapt the acoustic models to the recognition environments for represent properly the corrupted speech.

The concepts of these two categories can be described in a framework as presented in Figure 2.1 [13]. The feature space and the model space are presented in two sides respectively. For a particular attribute of speech signal, it can be distinguished in three levels. The first level lies in the training condition and the third level is located in the test condition. The second level is the intermediate canonical level, where the variations are removed. The mismatch can be determined in this framework by combination of the specific feature and model space condition. Mismatch exists if the acoustic feature and model do not belong to the same level. Otherwise there is no mismatch.

To reduce or overcome the mismatch, the normalization and adaptation schemes try to transform a specific acoustic feature and acoustic model into the same level, respectively. The normalization schemes transform the test features Y into the canonical form $\tilde{X}$. As the residual mismatch still exists between L_x and $\tilde{X}$, it is necessary to apply the normalization process on the training feature as well ($X \rightarrow \tilde{X}$). The adaptation techniques transform the acoustic model L_x into canonical level. With adaptation (e.g. Maximum Likelihood Linear Regression, MLLR) the acoustic model can also be transformed directly to a specific test condition, i.e., from L_x to L_y. This paper will focus on feature space normalization techniques subsequently.

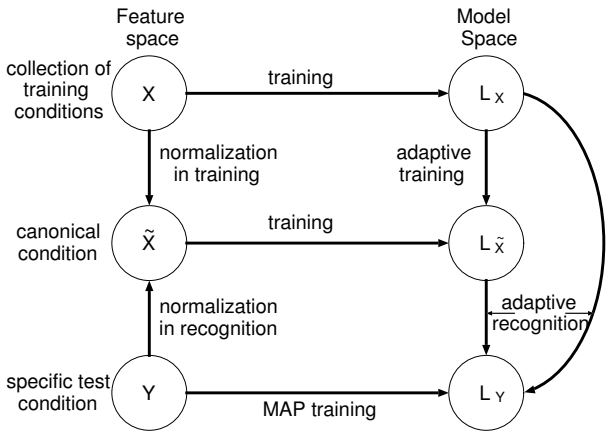

Figure 2.1: *Overview of normalization and adaptation concepts*

2.2 Normalization Techniques

Nowadays most of ASR systems use MFCC features as their front ends. The feature space normalization techniques discussed in this paper are only for MFCC front ends. Normalization aims at compensating the distortions caused by environment variations. As the variations perform either linear or non-linear effects, the compensation mechanisms are classified into two types: linear and non-linear transformation [2]. Cepstral Mean Normalization (CMN) and the combination with a normalization of the variance (Mean and Variance Normalization (MVN)) are well-known methods of linear transformation, while histogram normalization (HN) and quantile equalization (QE), which will be discussed in the following chapters, belong to the non-linear category.

Cepstral Mean and Variance Normalization

Every stage of the transmission has channel characteristics. Disturbance often occurs during transmission. Cepstral Mean Normalization (CMN) is desired to overcome the speech quality's degradation during transmission. It is a simple and effective way of normalization by subtracting the mean of each cepstrum coefficient to remove time-invariant distortions introduced by the transmission channel and recording device [12]. Further normalization of the variance of cepstral coefficients (CVN) or MVN helps improve recognition performance in adverse conditions [12]. Nevertheless, these linear transformation methods cannot compensate non-linear distortion caused by noise.

Histogram Normalization

In order to reduce the non-linear deterioration, the technique of histogram normalization, a non-linear unsupervised normalization technique for robust speech recognition [4] was introduced. This normalization approach focuses on matching the overall distribution of each feature space dimension in training and test collection, not just the mean and variance [12]. Hence it can generate good results. Following the paper will discuss it in detail, show how efficient it performs with combination of other normalizations and how to optimize it in implementation.

Chapter 3

Histogram Normalization Technique

Histogram based normalization (HN), also known as histogram equalization, was originally utilized for Digital Image Processing [5][19] to adjust the brightness and contrast of images. Recently this mechanism was introduced in ASR and the performance improvements have been reached [1][4]. In present this technique is further developed by RWTH Aachen research group [8][9] and other organizations [2][3], and is widely adapted to the speech representation in feature extraction of ASR.

3.1 Basic Algorithm

From a statistics viewpoint, mismatch between training and test conditions means the mismatch of distributions of the feature vectors' attributes. The main idea of HN is to remove undesired variations by transforming the speech signals to be recognized in such a way that their distribution matches that of the training data. To find out such a multi-dimensional transformation for this objective is difficult in general. But under the simplifying assumption of independence between the acoustic vector dimensions, distribution matching can be performed for each dimension separately by using one-dimensional density matching [4][11].

Let X represent clean training data samples and Y denote corrupted test data samples in a dimension. A mapping function is introduced to estimate the non-linear process between the distributions of them:

$$Y = f(X) \tag{3.1}$$

Here the non-linear mapping is restricted to be single valued and monotonically increasing in the interval [0,1] [18]. Let $p_X(x)$ and $p_Y(y)$ denotes the probability density functions (PDF) of the training and test data samples respectively. Their corresponding cumulative density functions (CDF) $P_X(x)$ and $P_Y(y)$ for any training data x and corresponding test data $y = f(x)$ can be described as:

$$P_X(x) = \int_{-\infty}^{x} p_X(u)du \tag{3.2}$$

$$P_Y(y) = \int_{-\infty}^{y} p_Y(v)dv \tag{3.3}$$

It is obvious that the distributions of two data samples are the same if they have the same CDFs. Therefore, by setting the CDF $P_Y(y)$ equal to the CDF $P_X(x)$, namely CDF matching, the single valued and monotonically increasing non-linear transformation can be identified. That is,

$$P_Y(y) = P_X(x) \tag{3.4}$$

$$x = P_X^{-1}(P_Y(y)) \tag{3.5}$$

where $P_X^{-1}(.)$ is the inverse function of the cumulative density function $P_X(x)$.

The Equations (3.4) and (3.5) are the practical expression of the desired transformation (mapping function). According to the Equation (3.5), any test data value y (unnormalized) can be transformed to the corresponding training data value x (normalized) based on CDF matching. The set of all x and y obtained in this manner would form the lookup table that represents the non-linear transformation process. This way, a clean data distribution can be estimated from that of the corrupted test utterances, the undesired effects from variations could be removed or reduced. This mapping process can be presented in Figure 3.1 [13].

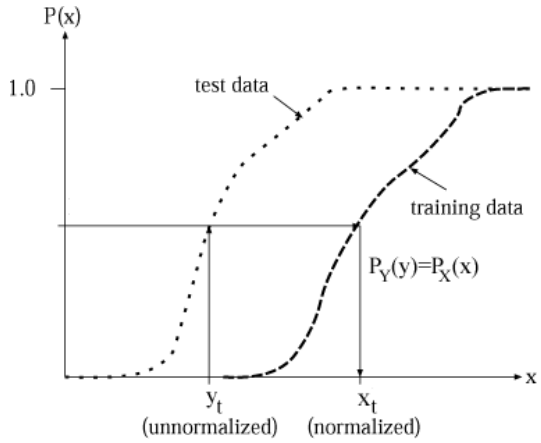

Figure 3.1: *Principle of histogram normalization*

In practice as the number of data samples is finite, distribution histogram and cumulative histograms are utilized to approximate the PDF and CDF respectively, this method is thus called histogram normalization. These approximations are described below [4].

Let x_{max} and x_{min} be the maximum and the minimum value of the data samples X for each dimension, respectively. The range from x_{min} to x_{max} are divided uniformly into M (typically about 10,000) non-overlapping intervals or bins, i.e.,

$$x_{min} = b_1 < b_2 < \cdots < b_{m+1} = x_{max} \quad \text{and bin} \quad B_i = [b_i, b_{i+1}) \tag{3.6}$$

Let N be the total number of data samples. Count the number n_i of data values that fall into any bin B_i through the entire data sample set. The PDF $p(x_i)$ of data values x being in bin B_i can be approximately computed by:

$$p(x_i) = p(x \in B_i) = \frac{n_i}{N} \tag{3.7}$$

Then the corresponding CDF $P(x_i)$ can be calculated by:

$$P(x_i) = P(x \leq x_i) = \sum_{j=1}^{i} \frac{n_j}{N} \tag{3.8}$$

which is a piece-wise constant function approximation of the true cumulative density function.

3.2 Practical Implementation

3.2.1 Computational Procedure

HN relies on the assumption that the process, which is responsible for the mismatch, has an independent effect on the different acoustic vector components [4][11]. Under this assumption the computing process of HN for each dimension can be described as follows:

1. Compute the PDF (distribution histogram) and the corresponding CDF $P_X(x_i)$ (cumulative histogram) of training data according to Equations (3.7) and (3.8).

2. Compute the PDF (distribution histogram) and the corresponding CDF $P_Y(y_i)$ (cumulative histogram) of recognition (test) data in the same way as in step 1.

3. Construct two tables from all data pairs $(x_i, P_X(x_i))$ and $(y_i, P_Y(y_i))$. $P_X(x_i)$ and $P_Y(y_i)$ take on values in [0,1] in equal increments (typically 1000 increments).

4. Combine these two tables with condition $P_Y(y_i) = P_X(x_i)$ to form a mapping table (y_i, x_i).

5. For any acoustic feature value y the closest value in $\{y_i\}$ is found by a simple table lookup and then the corresponding clean value in $\{x_i\}$ is chosen by the mapping table (y_i, x_i) as the transformed (normalized) value of y.

3.2.2 Training Data Normalization

In the algorithm described above, the distribution of training data is taken as the reference (target histogram) and the recognition data are normalized accordingly. As mentioned before in the framework of normalization and adaptation approaches (Figure 2.1), there still exists a mismatch between the normalized recognition feature $\tilde{X}$ and the training model L_X. For further mismatch reduction both training and recognition data need to be normalized [13].

First the overall distribution of all training data for each dimension is computed and the corresponding cumulative histogram is used as the target histogram. Then the distribution of training data from each speaker in each dialog is computed and transformed to match the target histogram. The recognition data are transformed in the same way.

3.2.3 Silence Fraction Treatment

Different silence fractions depending on the recording may cause degradation of histogram normalization [11][12]. With a much lower than average silence fraction inside recognized speech HN may transform some acoustic speech feature vectors to silence and cause word deletions. With a much high silence fraction some silence vectors will be transformed into speech and cause word insertions.

As depicted in Figure 3.2, the histogram is actually a manifestation of speech and silence in form of a bimodal structure [11]. In order to adapt the silence fractions by

different speakers, the original histogram can be split into silence and speech histograms. For the training speakers, a forced alignment with the reference transcriptions is proposed to map acoustic speech vectors and silence portions to two independent target histograms respectively. Then an adapted target histogram can be calculated by the cumulative speech histogram P_{speech} and cumulative silence histogram P_{sil} with an estimated silence fraction α of each speaker.

$$P_\alpha = \alpha \cdot P_{sil} + (1 - \alpha) \cdot P_{speech} \tag{3.9}$$

The adapted target histogram P_α is then used as usual for the basic algorithm of histogram normalization.

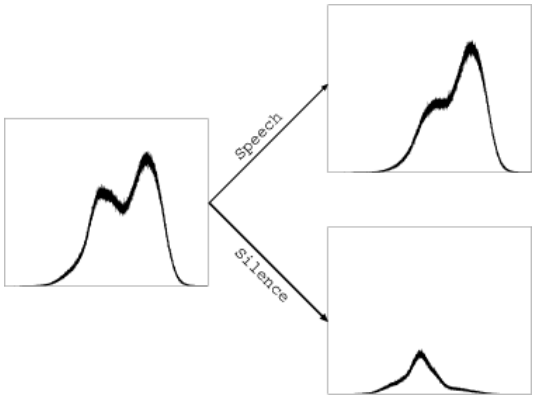

Figure 3.2: *Histogram over the third log filter bank coefficient on the VerbMobil II corpus [11]. The left side shows the original histogram, on the right side the histogram split into speech and silence.*

3.3 Experimental Evaluation

3.3.1 Applying HN at Different Feature Extraction Stages

In principle histogram normalization can be applied at any stage of MFCC feature extraction. Taking the characteristics of processing in different phrases and practical implementation into account, it may be applied at three stages [13], that is, after Logarithm, after Cepstral Mean Normalization (CMN) and after Linear Discriminant Analysis (LDA), as illustrated in Figure 3.3. After Mel-scaled filtering and Logarithm, the speech signal vectors leave in the order of 20 dimensions, which simplifies the histogram normalization. The spectral log-compression will also help to keep histogram discretization errors small in practice [13]. CMN is typically used to remove time-invariant distortions. It also helps to transform cepstral coefficients to unity variance. Apply HN to LDA transformed vectors intends to normalize the distributions of recognition vectors to those observed in training.

The task is to find out a reasonable way, at which stage or in which combination of three stages, for applying HN to obtain the best recognition performance. The tests are carried out on VerbMobil II corpus [13] with the RWTH large vocabulary continuous speech recognition system (RWTH LVCSR) by Informatik VI in the University of Technology Aachen (RWTH i6) [15].

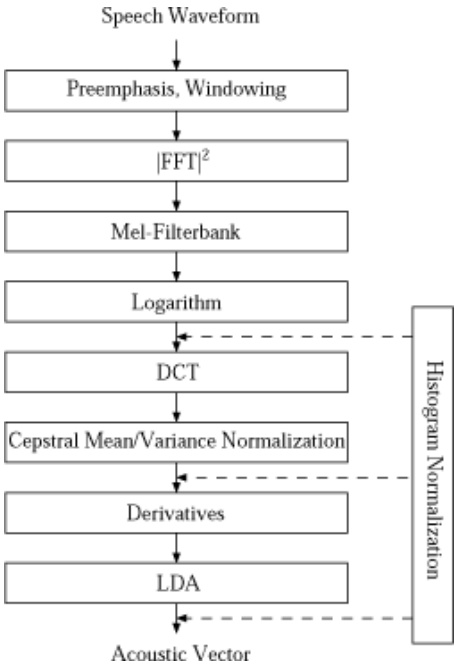

Figure 3.3: *Application of HN at different stages of feature extraction*

VerbMobil II [13] is a German spontaneous speech task with a 10k-word vocabulary. The statistics of training and test corpus are listed in Table 3.1. One part of the training data (49h) was collected with a headset microphone, the other (12h) with a room microphone. The training and test data were collected at different sites, so there are minor differences in the recording environments. The recognizer uses 16 cepstral features with first derivatives and the second derivatives of the energy, includes cepstral mean and variance normalization and LDA transformation, contains 2500 decision tree based within-word Triphone states including noise plus one state of silence and uses gender independent acoustic models with globally pooled diagonal covariance.

Table 3.1: *Statistics of the training and recognition corpus on VerbMobil II*

VerbMobil II		
	Training CD1-41	Test DEV99B
Duration [h]	61.5	0.5
Silence Portion [%]	13	11
# Speakers	857	6
# Sentences	36,015	336
# Words	701,512	4,346
Trigram PP.	-	74.6

The proposed method has been applied in two cases, namely in recognition only, and both in training and recognition. The test results are presented in Table 3.2. In both

conditions applying HN at different stages leads to error rate reduction up to 3%. It yields more gains at filterbank stage individually (23.8% word error rate, HN in recognition only) than at the other two, and further mismatch reduction (23.0%) can be obtained from normalized training data. The performance improvement of HN at different stages is to some extend additive. The best performance is obtained when applying HN both after Logarithm and after LDA (22.5%). But most gain is achieved by normalization at log-filterbank stage. Therefore, in later tests by RWTH i6 (Table 3.5 and 4.1) HN will usually be applied only after Logarithm.

Table 3.2: *Recognition results for HN at different feature extraction stages.*

VerbMobil II					
	Histogram Nomalization			Overall [%]	
	Filterbank	Cepstrum	LDA	Del - Ins	WER
in recognition only	No	No	No	4.9 - 4.4	24.6
	Yes	No	No	5.0 - 4.0	**23.8**
	No	Yes	No	4.5 - 4.3	24.0
	No	No	Yes	4.6 - 4.3	24.2
both in recognition and training	No	No	No	4.9 - 4.4	24.6
	Yes	No	No	4.9 - 4.4	**23.0**
	No	Yes	No	5.0 - 4.7	24.3
	No	No	Yes	4.9 - 4.4	24.1
	Yes	Yes	No	4.9 - 4.4	22.9
	Yes	No	Yes	4.3 - 4.0	**22.5**
	No	Yes	Yes	4.9 - 4.2	24.0
	Yes	Yes	Yes	4.9 - 4.3	22.7

3.3.2 Applying HN under Different Mismatch Conditions

The experiments to evaluate HN under different mismatch conditions have been carried out on Aurora-2 database by a research group in the University of Granada (Spain) [2][10].

Aurora-2 task consists of the recognition of connected digits spoken in US-English. It is based on a modified version of the TI-Digits database. Different noises are added to the speech at various signal-to-noise ratios. Two training sets (clean sets and multi-condition sets) and three test sets (set A, B, and C) are defined. Clean training set uses only clean digits while multi-condition set adds four different noises. Test set A has the same noises as the multi-condition training set, which results in high matched test condition. Test set B contains four different noises, which causes a mismatch between the training and test data. In set C, noisy speech is filtered to simulate channel mismatch.

The speech recognizer, HTK speech recognition toolkit defined for the ETSI Aurora evaluations [10], is based on Hidden Markov Modeling. Each digit is modeled as a left-to-right continuous density HMM with 16 states and six Gaussian per state. The front end is defined as original Aurora WI007 MFCC feature extraction [10]. The speech signal is represented as 39-dimentional feature vector containing a log-energy coefficient, 12 cepstral coefficients and the 1st and 2nd associated regression coefficients.

Different with the experiments performed by RWTH i6 (Table 3.5 and 4.1), in this

experiments the histogram normalization is applied in cepstral domain, not at the log-filterbank level. The target cumulative histogram is obtained from a Gaussian probability distribution with zero mean and unity variance.

The recognition results are summarized in Table 3.3. With multi-condition training mode the results present the same tendency of error rate reduction for three test sets, even their original mismatches are different. With clean training mode, in which there is a high mismatch of training and test data, the error rates are reduced drastically. The overall relative improvement in this task is 37.4%. That shows HN can significantly improve the recognition performance and make system more robust under different mismatch conditions.

Table 3.3: *Recognition results for HN on the Aurora-2 noisy TI digit strings. Multi: training with noise added at different SNRs. Clean: training without additional noise. Set A-C: different noise conditions for testing. Overall: 0.4 × Set A + 0.4 × Set B + 0.2 × Set C.*

Aurora-2 noisy TI digit strings					
WER [%] for MFCC baseline		Set A	Set B	SetC	Overall
	Multi	12.0	12.8	15.4	13.0
	Clean	41.3	46.6	34.0	41.9
	Average	26.6	29.7	24.7	**27.5**
WER [%] using HN	Multi	10.0	10.4	10.8	10.3
	Clean	19.7	18.6	19.6	19.2
	Average	14.9	14.5	15.2	**14.8**
Rel. Improv. [%]	Multi	17.8	24.2	33.3	23.5
	Clean	49.0	58.1	42.0	51.2
	Average	33.4	41.2	37.7	**37.4**

3.3.3 Combining HN with Other Compensation Methods

HN can be combined with other Compensation Methods to obtain further performance improvement. The experiments have been carried out on CarNavigation corpus with different combinations by RWTH i6 [12].

CarNavigation [12] is a German isolated word database with a 2k-word closed vocabulary and strong mismatch conditions. Training data were recorded in a quiet office environment (SNR 21dB). The office test set was recorded under the same situations. Two other test sets were recorded in driving cars (city and highway traffic, average SNRs 9dB and 6dB, respectively, microphone on the visor). There was no overlap in vocabulary between the different test sets, and between the training and test sets. Turn duration gives the average amount of acoustic data used to estimate the histograms in offline mode. Statistics of the training and test corpora are summarized in the Table 3.4. The recognizer, RWTH LVCSR [15], contains a standard MFCC front-end.

Table 3.4: *Statistics of the training and test corpus on CarNavigation. Each word was uttered once and the vocabulary was different for all four corpora.*

CarNavigation				
	Training	Test		
	Office	Office	City	High.
Duration [h]	18.8	1.7	1.7	1.8
Silence Portion [%]	60	69	73	75
Turn Duration [s]	785	425	450	468
# Speakers	86	14	14	14
# Words	61,742	2,069	2,100	2,100
Zerogram PP.	-	2,100	2,100	2,100

The partial test results are summarized in Table 3.5. Without any normalization the recognition procedure acts well in quiet office, whereas the performance is poor in noisy environments (#0). The CMN methods, which are usually applied to remove the global shift of the distributions, halve the error rate in a city traffic conditions, but a high number of error rate still remains in more noisy highway environment (#1). When the variance is normalized as well, the error rate under the same situation still stays in a higher level of 40% (#2). The combination of CMN and HN, however, can provide much more improvements (#3). Applying HN with explicit silence fraction treatment (#4) can obtain further error rate reductions of 7% to 20% relative to basic histogram normalization (#3).

Table 3.5: *Recognition results on CarNavigation. Cepstral mean normalization (CMN), cepstral variance normalization (CVN), histogram normalization (HN) with silence fraction treatment (HNSIL).*

CarNavigation				
#	Normalization	WER [%]		
		Office	City	High.
0	no normalization	2.8	68.0	99.0
1	CMN	2.9	31.6	74.2
2	CMN, CVN	4.2	20.8	39.7
3	CMN, HN	2.8	10.2	16.6
4	CMN, HNSIL	**2.6**	**8.2**	**14.3**

These experiments also include the tests with other combinations of HN, e.g. with feature space rotation and vocal tract length normalization [12]. The results are satisfactory as well. Experiments for HN carried out by S. Dharanipragada et al [4] show that it performs as well as the more computationally expensive MLLR technique and adds to the improvement when combined with MLLR. Other experiments with combinations of HN and spectral subtraction [17] or vector Taylor series approach [18] have shown that HN can efficiently remove the residual mismatch and recognition accuracy can be further improved.

3.4 Further Discussion

It is proved that the representations of speech signals are often suffered by an additive noise and/or a convolutional noise [2]. In the MFCC front ends, these two types of noise can be presented by using a simple model.

After Mel scaled filtering, the energy of the contaminated speech $Y_k(t)$ for the frame t and filter k, can be written as [2],

$$Y_k(t) = X_k(t) + N_k(t) \tag{3.10}$$

where $X_k(t)$ denotes the energies of the clean speech and $N_k(t)$ stands for the noise. If the signal is also affected by a convolutional noise, it follows,

$$Y_k(t) = (X_k(t) + N_k(t)) \cdot H_k(t) \tag{3.11}$$

Here the convolutional noise is characterized by $H_k(t)$ for the frequency band k. After applying the logarithm ($x_k = \log(X_k)$), this relationship is given by,

$$y_k(t) = h_k(t) + \log[\exp(x_k(t)) + \exp(n_k(t))] \tag{3.12}$$

In this domain, the convolutional noise causes a global shift of the parameters representing the speech, while the additive noise introduces a non-linear transformation on the feature space.

During feature extraction the CMN method is usually applied to remove the global shift of the mean in the cepstral domain. But this method is a linear approach that cannot compensate the non-linear effects by the noise. On the contrary, the HN methods provide more effective improvements. Since the formulation of the HN does not make any assumption about contamination process, they can compensate both linear and non-linear distortions of feature space. This can be proved from the test results in the Table 3.5 #1 and #3. After apply the HN, the WERs are reduced significantly by 67.7% relative on city data and 77.5% relative on highway data. In another experiment using the Aurora-2 database and task [2], similar results have been obtained.

From experimental results under various conditions on different databases, histogram based normalization is turned out to be an effective mechanism on reducing the mismatch between the training and test data. The processing of this method does not depend on the procedure affected by the variations and not on the representations of the speech signal, thus it can be widely used under different conditions in the feature extraction.

The effect of different compensation methods is to some extent additive. HN also performs a great achievement for residual noise compensation left by other normalization methods, such as CMN and CVN. Hence the combination of HN with other compensation methods like CMN is more competitive for mismatch reduction than applying those methods separately.

Histogram normalization relies on a proper estimation of the original data distribution by histograms. The more data the method can use to calculate the histograms, the more accurately the recognizer performs. In practice several minutes of speech signals have to been recorded for the computation. For this reason it is not suitable to apply the HN for online (real-time) recognition that only a short delay is permitted and only a small number of data samples (in a few seconds) can be observed. The solution for this case is to employ a parametric transformation. It is the main motivation of the quantile based histogram equalization discussed in the following chapter.

HN works on the assumption of independence between the feature vector dimensions. But this assumption does not always hold perfectly in real applications. HN can reduce the mismatch significantly but not completely. The combination of HN with other compensation methods is needed.

Chapter 4

Quantile Based Histogram Equalization

Quantile based histogram equalization (quantile equalization, QE) is a parametric type of histogram normalization for on-line applications where only a short delay is allowed and only short utterances are processed. It utilizes CDF matching method to make the CDFs of the recognition signal's value match that of the training data. The main point of the algorithm is to approximate the CDF with a small number (usually fewer than 10, typically 4) of quantiles instead of the full histogram, and then a parametric transformation function is applied to minimize the mismatch between the training and recognition quantiles.

4.1 Basic Algorithm

To apply the proposed method, individual quantiles of each feature vector component in the training and recognition conditions have to be determined and an appropriate transformation function has to be chosen.

By the mathematical definition of quantile, the p-quantile of CDF $P(y)$ is the number y_p such that $P(y_p) = p$. If P^{-1} is the inverse function of P, then $y_p = P^{-1}(p)$. For instance, if the number of quantiles N_Q is four, the CDF will be divided into four parts of 25% each and three quantiles Q_i $(i = 1, 2, 3)$ are located in the 25%, 50% and 75% position, respectively (Figure 4.1 [7]).

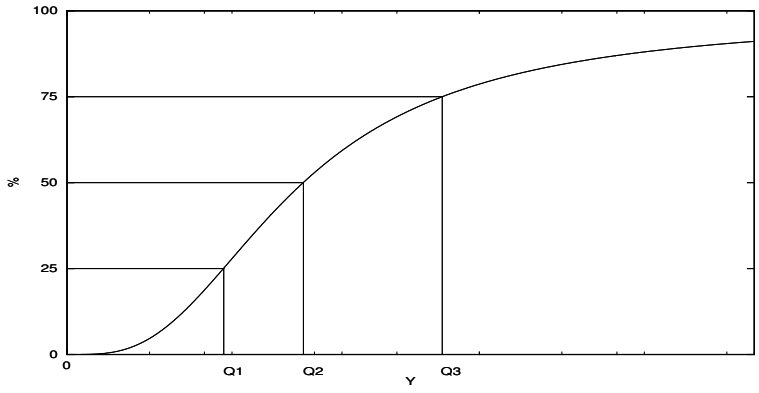

Figure 4.1: *A cumulative density function with three quantiles Q_i of 25% each*

In principle QE can be applied at any stage of the MFCC feature extraction [6]. Taking the position of the QE into account the transformation function can be chosen as linear, non-linear or other adequate function for reducing the mismatch. In [6] it is applied after Mel scaled filtering (Figure 4.2). The usually used logarithm is replaced by the 10th root compression for dynamic range reduction. As the piecewise linear transformation would have the problem on data with dynamic range for its discontinuous derivative [7], the transformation function used usually in practice is power function with an additional linear term.

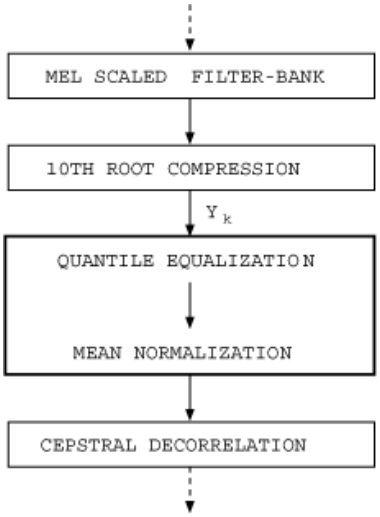

Figure 4.2: *Position of the quantile equalization*

Let N_Q be the number of quantiles, $Q_{k,i}$ and $Q_{k,i}^{train}$ represent the i-th recognition and training quantile for the k-th filter respectively, Q_i^{train} denote the average training quantiles over all filters and Q_{k,N_Q} stand for the maximal value of the recognition quantiles for the k-th filter. Let $Y_k[t]$ denote the output of the k-th Mel scaled filter. Before transformation the filter output values are scaled to the interval [0,1] to ensure transforming the values not closed to zero and high values. After transformation, the result values are scaled back to original range. The power transformation function T_k is described as:

$$T_k(Y_k[t]) = Q_{k,N_Q}\left(\alpha_k\left(\frac{Y_k[t]}{Q_{k,N_Q}}\right)^{\gamma_k} + (1-\alpha_k)\frac{Y_k[t]}{Q_{k,N_Q}}\right) \tag{4.1}$$

where the parameters α_k and γ_k are chosen to minimize the squared distance of the points $\{Q_{k,i}\ ,\ Q_i^{train}\}$ by:

$$\{\gamma_k, \alpha_k\} = \underset{\{\gamma_k,\alpha_k\}}{\operatorname{argmin}}\left(\sum_{i=1}^{N_Q-1}(T_k(Q_{k,i},\gamma_k,\alpha_k) - Q_i^{train})^2\right) \tag{4.2}$$

The average training quantiles Q_i^{train} are calculated over all filters,

$$Q_i^{train} = \frac{1}{K}\sum_{k=1}^{K} Q_{k,i}^{train} \tag{4.3}$$

where K is the number of filters. After finding out the optimal parameters α_k and γ_k by using a grid search, the transformation is applied to all the filter output values (Figure 4.3).

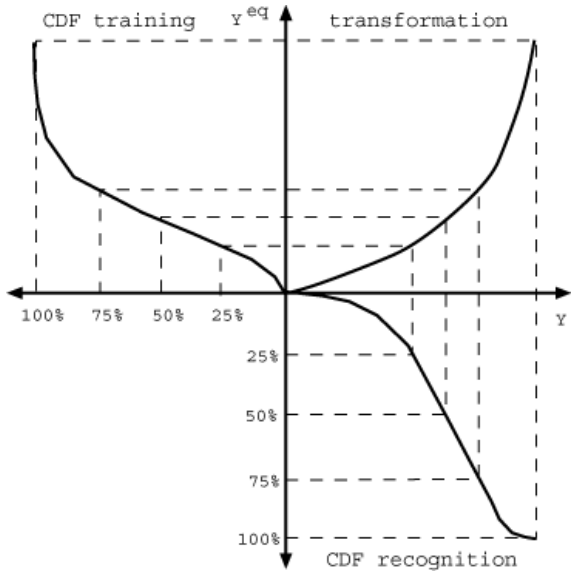

Figure 4.3: *Applying a transformation function to make the four training and recognition quantiles match.*

4.2 Online Implementation

4.2.1 Optimizing Parameters in Power Function Transformation

To avoid the scaling up of noise the $Q_{k,i}$ is restricted by defining lower bounds [7]:

$$if\ Q_{k,i} < Q_i^{train}\ then\ Q_{k,i} = Q_i^{train} \tag{4.4}$$

The transformation parameters are in the range $\alpha_k \in [0, 1]$ and $\gamma_k \in [1, 3]$. When using a full grid search in every time frame, the way of updating the transformation parameters affects strongly the recognition performance. The initial values used in the first time frame are $\alpha_k = 0$ and $\gamma_k = 1$ (no transformation). To avoid the sudden changes of the transformation parameters and increasing error rates from one time frame to the next, the updated values are only searched within a small range (e.g., 0.01) from the previous ones [6].

4.2.2 Combining QE with Additional Mean Normalization

The transformation using QE can be combined with mean normalization without inducing additional delay by using a sliding window instead of the whole utterance [6] (Figure 4.4). The window length and delay should be determined considering the application permit. Typical values are located in the range of 1s to 5s and 10ms to 500ms respectively. The signal's quantiles $Q_{k,i}$ for each filter channel are counted within a sliding window around the current time frame. Then the transformation is applied to all feature vectors within

the window. Subsequently the mean subtraction method can be applied to get a vector with zeromean.

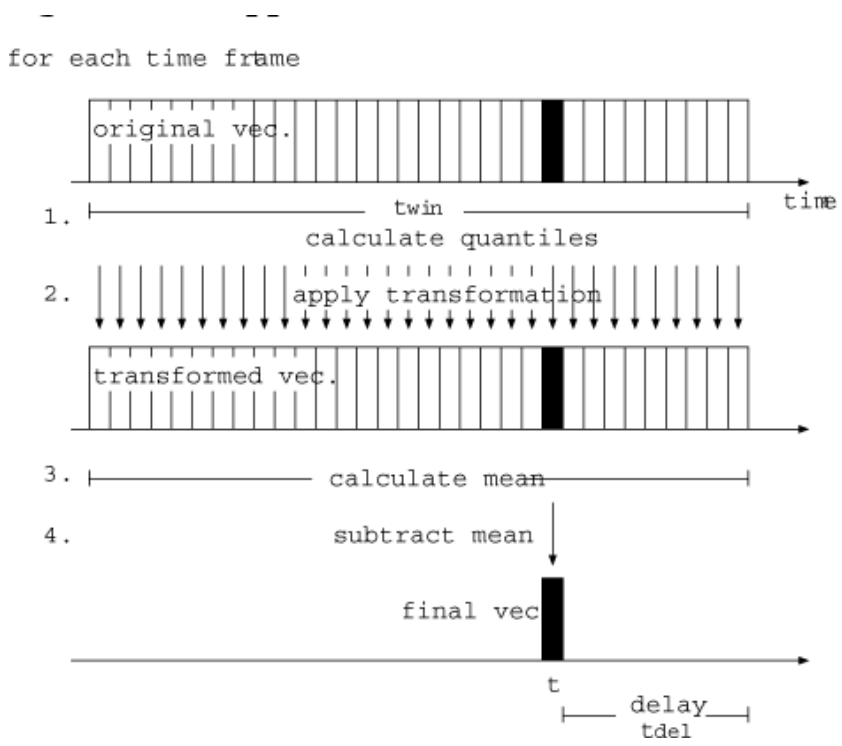

Figure 4.4: *Online normalizing scheme for the feature vectors after the Mel filter bank and 10th root compression.*

4.2.3 Training Data Normalization

Similar as in the HN, further improvements can be achieved when normalizing the training data as well. First all training data are used to estimate the target quantiles. Then the QE is applied on each training and recognition utterance to make its distributions match the target ones.

4.2.4 Computational Procedure

From the algorithm described above, the online computing process for each time frame can be summarized as follows:

1. Replace Logarithm by 10th root after Mel-scaled filtering.
2. Calculate the averaging training quantiles over all filters.
3. Calculate the signal's quantiles for each filter within the window around the current time frame.
4. Apply the power function transformation on all vectors within the window.
5. Calculate and subtract the mean of the transformed vectors.
6. Continue to calculate cepstral coefficients as usual using the result vectors.

4.3 Experimental Evaluation

4.3.1 Verification of the Algorithm

The algorithm of QE has been verified by several experiments on various databases [6][7][8][9]. The related test results showed that the optimal number of quantiles is four [7]. The recognition performance can be increased by averaging the training quantiles over all filters. Usually the power function transformation performs better than piecewise linear transformation [7].

The recognition tests on the CarNavigation database can be used to further evaluate the algorithm [16]. The database and recognizer setup for the tests are the same as described in Section 3.3.3. Test results are summarized in Table 4.1. Some HN test results are also included for comparison.

It turned out that replacing the logarithm by a 10th root compression alone (#3) can reduce the error rate by 40% relative to CMN (#1), and get the similar results as the CMN and CVN combination (#2). The normal QE leads to 50% further relative reductions (#4). Moreover, combining QE with mean normalization (FMN) does not induce additional delay. The total delay is 500ms with a window length of 1s to estimate the quantiles and the mean. The better results of up to 17% relative are obtained when training data are normalized as well (#5), and these results are similar to the results by HN alone(#6). That means enough recognition accuracy can be achieved by using the algorithm of QE with a short delay.

Table 4.1: *Recognition results on the CarNavigation corpus with different normalization techniques. Cepstral mean normalization (CMN), cepstral variance normalization (CVN), filter bank mean normalization (FMN), quantile equalization (QE) in training and test (QETT), histogram normalization (HN) with silence fraction treatment (HNSIL). Logarithm (log) and 10th root (root) were used to reduce the dynamic range of the filterbank channels.*

CarNavigation				
#	Normalization	WER [%]		
		Office	City	High.
0	log, no normalization	2.8	68.0	99.0
1	log, CMN	2.9	31.6	74.2
2	log, CMN, CVN	4.2	20.8	39.7
3	root, FMN	2.8	19.9	40.1
4	root, FMN, QE	3.2	11.7	20.1
5	root, FMN, QETT	**3.3**	**10.1**	**16.7**
6	log, CMN, HN	2.8	10.2	16.6
7	log, CMN, HNSIL	**2.6**	**8.2**	**14.3**

4.3.2 Applying QE on Different Mismatch Conditions

The recognition tests on Aurora-2 database [6] have been carried out in the similar manner as described in Section 3.3.2 to evaluate the effectiveness of quantile equalization method in different training mode and various noise environments.

The front end is a modified version of the original Aurora WI007 feature extraction, in which logarithm is replaced by 10th root compression, quantile equalization and mean normalization module is added between 10th root and the calculation of the cepstral coefficients, and 0th cepstral coefficient is used instead of log energy. The HTK speech recognition toolkit is used in the original setup defined for the ETSI Aurora evaluations [10].

The test results are also averaged for the three test sets (set A, B and C) considered in the Aurora-2 task, as shown in Table 4.2. With clean training mode there is high mismatch between training and recognition data, a large relative improvement of 50% can be achieved by QE. With multi-condition training mode the relative improvement is 15%. The overall average improvement on this database is 32%. The results for the three test sets have shown that error rate reduction can be obtained in different mismatch conditions. QE performs 5% less overall improvements than HN on this database (Table 4.2 vs. Table 3.3). The reason is that the former has been carried out in an online application whereas the latter can be performed in offline mode after the recording of entire utterances has been finished.

Table 4.2: *Recognition results for QE on the Aurora 2 noisy TI digit strings. Multi: training with noise added at different SNRs. Clean: training without additional noise. Set A-C: different noise conditions for testing. Overall: 0.4 × Set A + 0.4 × Set B + 0.2 × Set C.*

Aurora-2 noisy TI digit strings					
WER [%] for MFCC baseline		Set A	Set B	SetC	Overall
	Multi	12.0	12.8	15.4	13.0
	Clean	41.3	46.6	34.0	41.9
	Average	26.6	29.7	24.7	**27.5**
WER [%] using QE	Multi	10.2	10.8	10.8	10.5
	Clean	23.5	21.9	22.4	22.6
	Average	16.9	16.3	16.6	**16.6**
Rel. Improv. [%]	Multi	10.6	14.8	23.6	14.9
	Clean	43.4	59.9	42.0	49.7
	Average	27.0	37.4	32.8	**32.3**

4.4 Summary

Quantile based histogram equalization is a feature space normalization method especially for online speech recognition. The experiments have presented that a small amount of utterance is sufficient to estimate the cumulative density functions by using quantiles and error rate reduction can be obtained in different mismatch conditions. Using the 10th root compression and combining QE with mean normalization can significantly improve the recognition performance without additional delay. In noisy conditions the best results can be obtained when training data are also normalized. It shows explicitly that QE provides the recognition performance as well as the basic HN method. Thus QE can satisfy the demand of real-time applications.

Chapter 5

Conclusion

Acoustic variations usually cause a mismatch between training and recognition conditions and lead to a severe deterioration of recognition performances in speech recognition systems. Histogram normalization can be applied during feature extraction to reduce this mismatch significantly. It estimates the distribution of training and test data by using a histogram based approach and transforms the signals based on CDF (cumulative density function) matching. Quantile equalization is a parametric type of histogram normalization for online application. It approximates the CDFs with a small number of quantiles instead of the overall histogram, and a parametric transformation function is used to make the training and test quantiles match. The test results have shown that HN and QE are the effective methods to increase the robustness of speech recognition systems under different acoustic conditions. Normalization both on training and recognition data, and silence fraction treatment can make histogram normalization more effective. Quantile equalization with an additional mean normalization can further improve the overall performance even when the delay is short.

Histogram normalization is computationally inexpensive. Its processing does not depend on the procedure affected by the variations and not on the representations of the speech signal. Therefore it can be widely used under different conditions during feature extraction. Combined with other normalization techniques, e.g., Cepstral Mean Normalization, the further recognition performance improvement can be achieved.

Quantile equalization is improved in progress. A new approach, combining neighboring filter channels, has been proposed and evaluated for further improvement [8][9]. That shows histogram normalization techniques are competitive methods for reducing the mismatch and increasing the system's robustness.

Bibliography

[1] R. Balachandran and R. Mammone. Non-parametric Estimation and Correction of Non-linear Distortion in Speech Systems. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 749–752, Seattle, WA, USA, May 1998.

[2] A. de la Torre, J. C. Segura, C. Benitez, A. M. Peinado, and A. J. Rubio. Non-linear Transformations of the Feature Space for Robust Speech Recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 401–404, Orlando, Florida, USA, May 2002.

[3] Febe de Wet, Johan de Veth, Bert Cranen, and Louis Boves. The Impact of Spectral and Energy Mismatch on the AURORA2 Digit Recognition Task. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 105–108, Hong Kong, China, April 2003.

[4] S. Dharanipragada and M. Padmanabhan. A Nonlinear Unsupervised Adaptation Technique for Speech Recognition. In *Proc. Int. Conf. on Spoken Language Processing*, pages 556–559, Bejing, China, October 2000.

[5] R. C. Gonzalez and P. Wintz. *Digital Image Processing*, chapter 4, pages 173–178. Addison Wesley, 1987.

[6] F. Hilger, S. Molau, and H. Ney. Quantile Based Histogram Equalization for Online Applications. In *Proc. Int. Conf. on Spoken Language Processing*, volume 1, pages 237–240, Denver, CO, USA, September 2002.

[7] F. Hilger and H. Ney. Quantile Based Histogram Equalization for Noise Robust Speech Recognition. In *Proc. European Conference on Speech Communication and Technology*, volume 2, pages 1135–1138, Aalborg, Denmark, September 2001.

[8] F. Hilger and H. Ney. Evaluation of Quantile Based Histogram Equalization with Filter Combination on the Aurora 3 and 4 Databases. To appear in *Proc. European Conference on Speech Communication and Technology*, Geneva, Switzerland, September 2003.

[9] F. Hilger, H. Ney, O. Siohan, and F. K. Soong. Combining Neighboring Filter Channels to Improve Quantile Based Histogram Equalization. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 640–643, Hong Kong, China, April 2003.

[10] H.G. Hirsch and D. Pearce. The Aurora Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions. In *ASR2000*

- *International Workshop on Automatic Speech Recognition*, pages 181–188, Paris, France, September 2000.

[11] S. Molau, F. Hilger, D. Keysers, and H. Ney. Enhanced Histogram Normalization in the Acoustic Feature Space. In *Proc. Int. Conf. on Spoken Language Processing*, volume 1, pages 1421–1424, Denver, CO, USA, September 2002.

[12] S. Molau, F. Hilger, and H. Ney. Feature Space Normalization in Adverse Acoustic Condition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 656–659, Hong Kong, China, April 2003.

[13] S. Molau, M. Pitz, and H. Ney. Histogram Based Normalization in the Acoustic Feature Space. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, volume 1, pages 225–228, Madonna di Campiglio, Trento, Italy, December 2001. 4 pages, CD ROM, IEEE Catalog No. 01EX544.

[14] H. Ney. Lecture Notes of Speech Recognition. Lehrstuhl für Informatik VI in RWTH Aachen, 2001.

[15] H. Ney, L. Welling, S. Ortmanns, K. Beulen, and F. Wessel. The RWTH Large Vocabulary Continuous Speech Recognition System. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 853–856, Seattle, WA, USA, May 1998.

[16] L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*, chapter 3, page 70. Prentice-Hall, 1993.

[17] 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. Int. Conf. on Spoken Language Processing*, volume 1, pages 225–228, Denver, CO, USA, September 2002.

[18] J. C. Segura, M. C. Benitez, A. de la Torre, and A. J. Rubio. VTS Residual Noise Compensation. In *Proc. Int. Conf. on Spoken Language Processing*, volume 1, pages 409–412, Denver, CO, USA, September 2002.

[19] A. Torre, A. M. Peinado, J. C. Segura, J. L. Prez, C. Bentez, and A. J. Rubio. Histogram equalization of the speech representation for robust speech recognition. Submitted to *IEEE Transactions on Speech and Audio Processing*, October 2001.