JOINT UNCERTAINTY DECODING (JUD) WITH HISTOGRAM-BASED QUANTIZATION (HQ) FOR ROBUST AND/OR DISTRIBUTED SPEECH RECOGNITION

Chia-yu Wan and Lin-shan Lee
Graduate Institute of Communication Engineering, National Taiwan University, Taiwan, Republic of China
chiayui@speech.ee.ntu.edu.tw and lslee@gate.sinica.edu.tw

ABSTRACT

Histogram-based Quantization (HQ) has been recently proposed as a robust and scalable quantization approach for Distributed Speech Recognition (DSR). In this paper, Histogram-based Quantization (HQ) is further verified as an attractive feature transformation approach for robust speech recognition, Joint Uncertainty Decoding (JUD) is developed to be applied with HQ for improved recognition accuracy, and the approach was evaluated for both cases of robust speech recognition and DSR. In Joint Uncertainty Decoding (JUD), we jointly consider and estimate the uncertainty caused by both the environmental noise and the quantization errors in Viterbi decoding under the framework of HQ. For robust speech recognition, HQ was used as the front-end feature transformation and JUD as the enhancement approach at the back-end recognizer. For DSR, HQ was applied at the client end as a data compression process and JUD at the server. The evaluation with Aurora 2.0 testing environment showed very significant improvements for both cases of robust and/or distributed speech recognition.

1. INTRODUCTION

Various applications of the automatic speech recognition (ASR) technologies in the future have been highly anticipated by many people. But the recognition accuracy of ASR systems is very often seriously degraded by the mismatch between the training and testing environments, hence robustness for ASR technologies with respect to the environmental disturbance has always been a key issue in real applications.

On the other hand, the client-server framework for Distributed Speech Recognition (DSR) has been widely considered, in which speech features are extracted and compressed in the hand-held clients and the recognition performed at the server. A new concept of Histogram-based Quantization (HQ) was recently proposed to be performed at the client for feature compression [1], in which the partition cells are dynamically defined by the histogram or order statistics of a segment of most recent past samples of the parameter to be quantized. Many problems generated due to a fixed VQ codebook in conventional DSR are automatically solved with this new Histogram-based Quantization (HQ) approach, because no fixed codebook is used at all. The mismatch between the corrupted feature vectors and the fixed codebook does not exist any longer, and most of the disturbance can be absorbed by the dynamic histogram [1]. Below in this paper, it is further found that this Histogram-based Quantization (HQ) can also be used as a good approach for robust feature transformation as well.

For both the above cases of robust and/or distributed speech recognition, the feature vectors used by the recognizer corrupted by both the environmental noise and the quantization errors can be viewed as random vectors with uncertainty. Unlike the standard Viterbi decoding process in which such vectors are considered as deterministic, the uncertainty decoding approach considers the uncertainty of these random vectors [2,3,4,5]. Approaches for robust ASR have been modified in the past to estimate such uncertainty produced by the environmental noise [2,3]. Extended Cluster Information Vector Quantization (ECI-VQ) was also developed to estimate the uncertainty generated in the quantization process [4]. However, in such cases, it is actually better to jointly consider the uncertainty for the quantized feature vectors caused by both the environmental noise and the quantization errors. The difficulties are that the environmental noise are hidden in the quantized codewords, not only not easy to estimate, but mixed with the quantization errors. The mismatch between the noisy feature vectors and the fixed VQ codebook usually further degrades the recognition performance. As will be clear below, the recently proposed Histogram-based Quantization (HQ) approach is able to solve to a good degree the problems mentioned above [1]. The Histogram-based Quantization (HQ) is therefore taken as the fundamental scheme in this paper, on which the Joint Uncertainty Decoding (JUD) proposed here is based.

In this paper, we consider both cases of robust and/or distributed speech recognition. We jointly estimate the uncertainty caused by both the environmental noise and the quantization errors in an ASR system using Histogram-based Quantization (HQ), and perform the Joint Uncertainty Decoding (JUD) at the recognizer. For robust speech recognition, HQ is used as the front-end feature transformation and JUD as the enhancement approach at the back-end recognizer. For Distributed Speech Recognition (DSR), HQ is applied at the client end as a quantization process for data compression and JUD at the server.

2. HISTOGRAM-BASED QUANTIZATION (HQ)

Histogram Equalization (HEQ) has been proposed to equalize the cumulative distributions (or histograms) of both the training and testing feature parameters in each temporal span (an utterance or a moving segment of frames), and shown to produce much more robust features for recognition [6,7,8]. The recently proposed Histogram-based Quantization (HQ) actually borrowed this concept to perform feature parameter quantization for DSR purposes based on the dynamically changing cumulative distributions (or histograms) of the feature parameters instead of a fixed VQ codebook [1]. Below in this paper we show HQ can be used as a robust feature transformation approach as well.

2.1 General Formulation of HQ

The concept of HQ is to perform the quantization of a feature parameter x_t at time t based on the histogram or order statistics of that feature parameter within a moving segment of most recent past T samples, $[x_{t-T+1}, \cdots, x_{t-1}, x_t] \triangleq X_{t,T}$, up to the time t being considered. As shown in Fig. 1, the values of these T parameters in $X_{t,T}$ are sorted to produce a time-varying cumulative distribution function C(y), or histogram, where $C(y_0)=b_0=0$ and $C(y_N)=b_N=1$, y_0 and y_N are respectively the minimum and maximum values within $X_{t,T}$. The N quantization levels, $\{D_i, i=1,2, \cdots,N\}$ are defined on

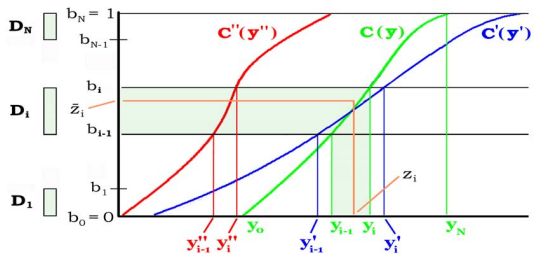

Fig. 1. Basic formulation of Histogram-based Quantization (HQ).

the vertical scale in [0,1] and then transformed to the range of the feature parameter on the horizontal scale, $[y_0, y_N]$, by $C^{-1}(y)$ constructed with $X_{t,T}$, to be the N partition cells for quantization of x_t. For example, the quantization level D_i, or $[b_{i-1}, b_i]$ on the vertical scale are transformed to the partition cell $[y_{i-1}, y_i]$ on the horizontal scale, where $C(y_{i-1})= b_{i-1}$, $C(y_i)= b_i$. In other words, the quantization here is a mapping relation, which maps the present parameter x_t being considered to a representative value z_i for the partition cell $[y_{i-1}, y_i]$ just as the conventional quantization process,

$$x_t \rightarrow z_i \text{ , if } b_{i-1} < C(x_t) < b_i, \quad \text{or } y_{i-1} < x_t < y_i, \ i=1, 2, \ldots, N. \tag{1}$$

The quantization levels D_i, together with its corresponding representative value $\bar{z}_i$ on the vertical scale are derived using a cumulative distribution $C_0(y)$ for a zero-mean standard Gaussian via the Lloyd-Max algorithm. The partition cell $[y_{i-1},y_i]$ on the horizontal scale is dynamic which is transformed from D_i by the time-varying histogram C(y), while the representative value z_i on the horizontal scale is fixed which is transformed from $\bar{z}_i$ by the standard histogram $C_0(y)$. So HQ is based on a hidden codebook $\{(D_i, \bar{z}_i), i=1,2, \cdots,N\}$ on the vertical scale, but transformed by dynamic histogram C(y) into time varying partition cells $[y_{i-1}, y_i]$ and by a fixed standard histogram $C_0(y)$ into the fixed representative value z_i on the horizontal scale. So the quantization codebook is not fixed any longer, and it has been shown that many problems with a fixed codebook in conventional DSR framework are automatically solved [1]. Note that here HQ is a quantization process, but can also be used as a feature transformation process for robustness process, in which each parameter x_t is transformed to its representative value z_i for the corresponding partition cell. This HQ approach has also been successfully extended to vector quantization and verified by very good performance [1].

2.2 Robust Nature of HQ

Conventionally, robust feature extraction and feature quantization are respectively for robustness and data compression purposes. HQ proposed here, however, automatically integrates the two different purposes. When a segment of parameters $X_{t,T}$ are corrupted by small disturbances, all individual values may be changed, but the order statistics which produces the partition cells on the horizontal scale may remain similar. For example, as shown in Fig. 1, with the disturbances, C(y) may be changed to C′(y′). The partition cell for the quantization level D_i=$[b_{i-1}, b_i]$ for the disturbed parameter x'_t may also be changed to $[y'_{i-1}, y'_i]$ where $C'(y'_{i-1}) = b_{i-1}$, $C'(y'_i)= b_i$, which can be quite different from $[y_{i-1}, y_i]$. However, the quantization level D_i and the corresponding representative value z_i for the disturbed parameter x'_t may remain unchanged as long as $y'_{i-1} < x'_t < y'_i$, since D_i is fixed on the vertical scale, while z_i is fixed on the horizontal scale. In other words, the quantization is based on the quantization levels D_i on the vertical scale and the histogram C(y), therefore less sensitive to the disturbances on the horizontal scale, or the disturbances on the horizontal scale is kind of "absorbed" by the dynamic histogram C(y). Although C(y) is disturbed into C′(y′), the changes on the quantization results may be very limited. Such robustness is obtained by the local order statistics for the parameters within the most recent values of the parameter, therefore it is dynamic and verified to be able to handle various noisy conditions including non-stationary disturbances [1].

3. JOINT UNCERTAINTY DECODING (JUD)

3.1 General Formulation of Uncertainty Observation Decoding

In standard HMM decoding, the observation probability $b_j(v)$ for observing a vector v at state j is

$$b_j(v) = \sum_{m=1}^{M} c_{jm} N(v; \mu_{jm}, \Sigma_{jm}), \tag{2}$$

where m is the mixture index, $c_{jm}, \mu_{jm}, \Sigma_{jm}$ are the mixture weight, mean, and covariance respectively for the m-th Gaussian mixture in state j. Uncertainty observation decoding considers the observation vector v in Eq.(2) as a random vector w with a distribution p(o|w), where o is a sample value of w. Eq.(2) is therefore extended to,

$$b_j(w) = \int_o p(o \mid w) b_j(o) do. \tag{3}$$

Assuming p(o|w) as a single Gaussian with mean μ_w, covariance Σ_w, $p(o \mid w) \sim N(\mu_w, \Sigma_w)$, the integration in Eq.(3) can be simplified to

$$b_j(w) = \sum_{m=1}^{M} c_{jm} N(\mu_{jm}, \Sigma_{jm} + \Sigma_w)\big|_{\mu_w}. \tag{4}$$

So the standard HMM decoding using Eq.(2) remains unchanged, except the variance of each Gaussian in the HMMs is increased by a quantity equal to Σ_w representing the uncertainty of the observation vector. In this way, the decoding can be more based on reliable parameters with smaller variance Σ_w.

3.2 Joint Uncertainty Decoding (JUD) for HQ

3.2.1 Uncertainty for quantization errors

In a HQ partition cell, the representative value z_i can be viewed as the mean value of a random variable. Σ_w in Eq.(4) can thus be estimated for a partition cell $[y_{i-1}, y_i]$ using a training set,

$$\Sigma_w^{q,i} = \frac{1}{N_i} \sum_{y_{i-1} < x_t < y_i} (C_0^{-1}[C(x_t)] - z_i)^2 \tag{5}$$

where the summation is over all parameters $y_{i-1} < x_t < y_i$ in the training set, and N_i is the total number of such parameters. Because the representative value z_i was obtained via Lloyd-Max algorithm based on the histogram $C_0(\cdot)$ for a standard Gaussian distribution, all parameters x_t in the partition cell need to be transformed back via $C_0^{-1}(\bullet)$ to evaluate $\Sigma_w^{q,i}$. Because the Lloyd-Max algorithm produces tightly quantized levels on high density region and loosely quantized levels on low density region to minimize the total distortion, the uncertainty observation decoding automatically increases the Gaussian variances for the loosely quantized levels.

3.2.2 Uncertainty for environmental noise

Under low SNR conditions, the disturbances may be very serious. For example in Fig. 1, y_{i-1} and y_i may be changed to y''_{i-1} and y''_i and C(y) to C″(y″), or there can be a histogram shift which can't be absorbed by the dynamic histogram. The performance of HQ is then inevitably deteriorated. Such a histogram shift may be reasonably estimated by $C^{-1}(0.5)$, because $C_0^{-1}(0.5)=0$ for a standard zero-mean Gaussian. These values of $|C^{-1}(0.5)|$ averaged over a training set for different SNR values were actually found to be roughly proportional to the noise variance. Therefore the

uncertainty caused by the environmental noise at time t can be estimated as

$$\Sigma_w^{n,t} = \alpha (C_t^{-1}(0.5))^2, \quad (6)$$

where $C_t(\cdot)$ is the histogram for the moving segment at frame index t, and α is a scaling factor determined empirically. This uncertainty is different for each frame t.

3.2.3 Joint uncertainty observation decoding

The above two types of uncertainty should be jointly considered. A reasonable assumption is that for high SNR conditions the uncertainty for quantization errors $\Sigma_w^{q,i}$ dominates, while for low SNR conditions that caused by environmental noise $\Sigma_w^{n,t}$ dominates. Therefore the joint uncertainty can be estimated as

$$\Sigma_w^{i,t} = \max(\Sigma_w^{q,i}, \Sigma_w^{n,t}), \quad (7)$$

where $\Sigma_w^{q,i}$ and $\Sigma_w^{n,t}$ are from Eq.(5) and (6). This value is different for different partition cell i and different frame t, and can be used directly in Eq.(4). Note that the uncertainty estimation here is based on HQ quantized features only. Therefore for DSR, it can be performed on the server easily without extra cost of bit rate.

3.3 Histogram Shift Compensation

As mentioned previously the histogram shift may seriously degrade the performance of HQ. In addition to the uncertainty decoding as mentioned above, we can also shift the histogram directly to have

$$C_t^{-1}(0.5) = 0 \quad (8)$$

for all frame t. Much of the serious disturbances can be absorbed by such a shift, as will be verified by the experiments below.

4. EXPERENMENTAL RESULTS

The experiments below were performed on the AURORA 2.0 testing environment for English digits strings. To evaluate the robustness against mismatched conditions, the clean-speech training condition with testing conditions sets A, B and C were tested with SNR ranging from 20 dB to 0 dB. The WI007 front-end was used to obtain 13 MFCC coefficients (C1~C12 and log energy) plus the delta and delta-delta features for recognition.

4.1 HQ and JUD for Robust Speech Recognition

In the first set of experiments, we considered the case of robust speech recognition, in which HQ as a feature transformation technique, i.e. each feature parameter x_t is transformed to the representative value z_i for the corresponding partition cell, and JUD was then performed during recognition to improve the performance. All the experiments reported here were based on order-statistics over segments of most recent past parameter values as mentioned in section 2, so there was no time delay. Better results were obtainable if such no-delay condition was removed, but left out here for space limitation.

The results are shown in Fig. 2(a), (b) and (c). The five bars in each set in the figure are respectively for the well-known histogram equalization (HEQ) alone, HQ alone, HQ with histogram shift (HQ-s, section 3.3), HQ with histogram shift plus uncertainty for environmental noise (HQ-s,n, sections 3.3 and 3.2.2) and HQ with hitogram shift plus uncertainty for environmental noise and quantization errors (HQ-s,n,q, sections 3.3 and 3.2), respectively averaged over all SNR values but separated for different types of noise (Fig. 2(a)), averaged over all different types of noise but separated for different SNR values (Fig. 2(b)), and averaged over all types of noise and all SNR values but separated for sets A, B, C (Fig. 2(c)). Here HEQ was equally performed with a moving segment of most recent T past parameters, and the same value of T=100 was used for all experiments for HEQ and HQ. It can be

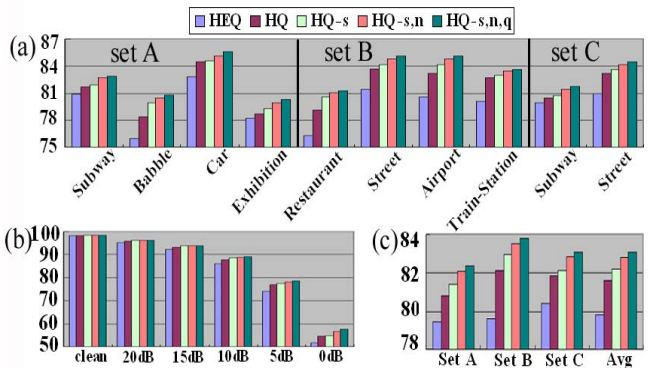

Fig. 2. Comparison of different approaches in this paper for robust speech recognition: (a) averaged over all SNR values but separated for different noise types in sets A, B, C, (b) averaged over all noise types but separated for each SNR value, and (c) averaged over all SNR values and noise types but separated for sets A, B, C.

Accuracy	Set A	Set B	Set C	Overall
HEQ	79.47	79.60	80.43	79.83
HQ-s,n,q	82.40	83.80	83.11	83.10
Relative error reduction (%)	14.30	20.59	13.73	16.23

Table 1. Exact accuracies and error rate reduction for HEQ and HQ-s,n,q for different testing sets in Fig. 2(c).

easily found that HQ consistently offered more robust features as compared to HEQ in all cases, and with the various approaches proposed here performed in the decoding process the performance can be consistently improved step by step as well. Also, the improvements were more significant for lower SNR cases (Fig. 2(b)), and for several types of non-stationary noise (Fig. 2(a)). Furthermore, the histogram shift compensation (HQ-s, 3rd bar) offered more improvements for non-stationary noise such as babble, restaurant and airport, than HQ alone (2nd bar, Fig. 2(a)), probably because for these types of noise the histogram shift was more significant. In addition, there was almost no performance degradation for clean or higher SNR conditions (Fig. 2(b)), apparently the uncertainty decoding for HQ is able to preserve the discrimination among HMMs. It is clear here that quantization process certainly produces quantization errors, but with proper design of the quantizer and the uncertainty decoding, quantization errors and environmental disturbances can in fact be well absorbed and compensated for. Exact accuracies for two cases in Fig. 2(c) (HQ-s,n,q and HEQ) are also compared in Table 1. Significant error rate reduction was achieved in all testing sets.

It is important to explain why HQ offered more robust features than the popularly used approach of histogram equalization (HEQ), and how HQ is different from the quantile-based HEQ [8]. HEQ performed point-to-point transformation based on the order-statistics or histogram, which can "absorb" the small disturbances to a good degree, although some "residual disturbances" inevitably remain because the point-based order-statistics is in any case more or less disturbed. The quantile-based HEQ performed a piecewise-linear approximation of HEQ. It reduced the computation complexity for histogram estimation, but didn't change the point-based nature of the transformation. HQ, on the other hand, performed the transformation by block-based order-statistics, therefore the small disturbances within a block (D_i in Fig. 1) were automatically absorbed. The block-based order-statistics certainly introduced uncertainty as well, but with proper choice of the number of quantization levels N or the block size, the uncertainty

may be reasonably taken care of by the stochastic nature of the Gaussian mixtures in the HMMs, and the uncertainty decoding can offer extra compensation in addition. For example, in all cases in Fig. 2, scalar HQ was always performed with N=8 blocks for all parameters and all SNR values. N=8 blocks seemed to be quite coarse for feature representation, but the results here indicated that the coarse blocks in fact brought robustness, because the small disturbances within the blocks were absorbed as mentioned above. Of course, it is natural that the best value of N should be SNR dependent, for example smaller N is better for lower SNR, and Histogram-based Vector Quantization (HVQ) is certainly even better [1], but such discussions are left out here for space limitation. At least N=8 for all cases for scalar HQ already verified the points mentioned here. As in Fig. 2(a), HQ alone turned out to be very helpful for babble/restaurant noise as compared to HEQ alone, probably because in such cases very often the disturbances of order statistics were absorbed by the blocks. For subway noise, on the other hand, the improvements of HQ alone compared to HEQ are relatively less, probably because the impulse-like disturbances may very often exceed beyond the blocks for N=8 here.

We further compared HEQ with HQ using a different metric, the averaged normalized distance between the corrupted features y_t and the corresponding clean speech features x_t,

$$d = \frac{1}{\sigma} E[y_t - x_t], \tag{9}$$

where the average is over all feature parameters in all the testing speech in sets A, B, C, and σ is the standard deviation of all the clean features x_t. Smaller values of d apparently imply the features are less influenced by the noise. The results are listed in Table 2 for different SNR values. We found in the table that HQ gives smaller values of d in all cases.

4.2 HQ and JUD for Distributed Speech Recognition (DSR)

For DSR, HQ was applied at the client end to replace the conventional Split Vector Quantization (SVQ) [9] for feature compression, and JUD performed at the server to improve the recognition performance. The results for this set of experiments are in Fig. 3(a)(b) and (c). The six bars in each set in the figure are respectively for the conventional Split Vector Quantization (SVQ), the Extended Cluster Information Vector Quantization (ECIVQ) [3], the cascade of HEQ and SVQ (HEQ-SVQ), HQ plus JUD considering both environmental noise and quantization errors (HQ-n,q), and HQ with JUD plus histogram shift (HQ-s,n,q). Fig. 3(a) are those averaged over all SNR values but separated for different noise types in sets A, B, C, (b) are those averaged over all types of noise but separated for different SNR value, while (c) are those averages over all types of noise and all SNR values but separated for sets A, B, C. In each case the bit rate required for the conventional Split Vector Quantization (SVQ) is 4.4 kbps and for HQ (all scalar with N=8 and T=100) is 3.9 kbps [1]. We can find that ECIVQ performed better than SVQ for sets A, B but slightly worse for set C, probably because ECIVQ considered quantization noise only, but the channel mismatch for set C might move the feature vectors to different partition cells, for which the cluster variance was not able to help. HEQ offered very good improvements to SVQ (HEQ-SVQ), but HQ proposed here consistently improved the performance more in almost all cases, and JUD (HQ-n,q) and histogram shift (HQ-s,n,q) further offered additional improvements consistently in almost all cases. The exact accuracies for HEQ followed by SVQ (HEQ-SVQ) and HQ-s,n,q are compared in Table 3. The improvements were significant and consistent for all SNR values, including for clean and 20 dB cases.

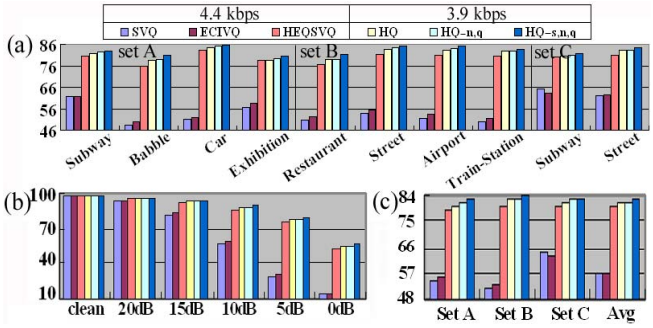

Fig. 3. Comparison of different approaches discussed in this paper for distributed speech recognition: (a) averaged over all SNR values but separated for different noise types in sets A, B, C, (b) averaged over all noise types but separated for different SNR values, and (c) averaged over all SNR values and noise types but separated for sets A, B, C.

SNR	20dB	15dB	10dB	5dB	0dB	-5 dB
HEQ	0.7876	0.8695	0.9516	1.0384	1.1314	1.2276
HQ	0.7172	0.7870	0.8588	0.9362	1.0204	1.1087

Table 2. The averaged normalized distance between clean and corrupted speech features under different SNR for HEQ and HQ

SNR	Clean	20dB	15dB	10dB	5dB	0dB
HEQ-SVQ	98.07	94.95	91.97	85.86	74.45	52.02
HQ-s,n,q	98.48	96.27	93.90	89.20	78.57	57.58
Error reduction (%)	21.28	26.18	24.02	23.64	16.12	11.58

Table 3. Exact accuracies and error rate reduction for HEQ-SVQ and HQ-s,n,q for different SNR values in Fig. 3(b).

5. CONCLUSIONS

Joint Uncertainty Decoding (JUD) under the framework of Histogram-based Quantization (HQ) is proposed here in this paper for robust and/or distributed speech recognition. Improved recognition performance was obtained consistently under all types of noise at all SNR values.

6. REFERENCES

[1] Chia-yu Wan and Lin-shan Lee, "Histogram-based Quantization (HQ) for Robust and Scalable Distributed Speech Recognition", Eurospeech2005.

[2] Jasha Droppo, Alex Acero,and Li Deng,"Uncertainty Decoding with SPLICE for Noise Robust Speech Recognition",ICASSP2002

[3] Jon A. Arrowood and Mark A. Clements, "Using Observation Uncertainty In HMM Decoding", ICSLP2002.

[4] Jon A. Arrowood and Mark A. Clements, "Extended Cluster Information Vector Quantization (ECI-VQ) for Robust Classification," ICASSP2004.

[5] H. Liao and M. J. F. Gales, "Joint Uncertainty Decoding for Noise Robust Speech Recognition," Eurospeech2005.

[6] S. Molau, M. Pitz, and H. Ney, "Histogram based normalization in the acoustic feature space, " ASRU2001.

[7] Á. de la Torre, J. C. Segura, C. Benítez, A. M. Peinado, and A. J. Rubio, "Non-linear Transformations of the Feature Space for Robust Speech Recognition", ICASSP 2002.

[8] F. Hilger and H. Ney, "Quantile-based histogram equalization for noise robust speech recognition," EuroSpeech 2001.

[9] "Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Advanced front-end feature extraction algorithm; Compression algorithms," ETSI ES 202 050 v1.1.3, ETSI standard, 2003.