Histogram-Based Quantization for Robust and/or Distributed Speech Recognition

Chia-Yu Wan, *Student Member, IEEE*, and Lin-Shan Lee, *Fellow, IEEE*

***Abstract*—In a distributed speech recognition (DSR) framework, the speech features are quantized and compressed at the client and recognized at the server. However, recognition accuracy is degraded by environmental noise at the input, quantization distortion, and transmission errors. In this paper, histogram-based quantization (HQ) is proposed, in which the partition cells for quantization are dynamically defined by the histogram or order statistics of a segment of the most recent past values of the parameter to be quantized. This scheme is shown to be able to solve to a good degree many problems related to DSR. A joint uncertainty decoding (JUD) approach is further developed to consider the uncertainty caused by both environmental noise and quantization errors. A three-stage error concealment (EC) framework is also developed to handle transmission errors. The proposed HQ is shown to be an attractive feature transformation approach for robust speech recognition outside of a DSR environment as well. All the claims have been verified by experiments using the Aurora 2 testing environment, and significant performance improvements for both robust and/or distributed speech recognition over conventional approaches have been achieved.**

***Index Terms*—Error compensation, robustness, speech recognition, vector quantization (VQ).**

I. Introduction

A WIDE variety of potential applications for automatic speech recognition (ASR) technologies have been highly anticipated. However, the recognition accuracy of ASR systems is always the core concern, which is very often seriously degraded by the mismatch between training and testing environments. Hence, robustness for ASR technologies with respect to environmental disturbances is definitely a key issue when considering real-world applications.

In addition, the client-server framework for distributed speech recognition (DSR) has been widely accepted, in which speech features are extracted and compressed at hand-held clients and recognition is performed at the server [1]. Various schemes for compression of ASR features have been proposed in recent years. Distance-based vector quantization (VQ) has been found very useful for clean speech and/or matched VQ codebook conditions [2], [3] and split vector quantization (SVQ) has been recommended by the ETSI standard [4]. However, environmental noise and quantization distortion naturally tend to jointly degrade recognition performance. The quantization process may increase the distance between clean and noisy features, and environmental noise may also move the feature vectors to a different quantization cell. The quantization distortion is actually related to the bit rates, which is another key parameter in DSR. The higher bit rate required for lower quantization distortion naturally becomes another difficult issue for transmission. Vector quantization or SVQ performed in a transformed domain (obtained with transforms such as discrete cosine transform (DCT) [5]–[7] or histogram equalization (HEQ) [8]–[10]) has been shown to be able to efficiently improve the desired robustness for feature vectors under environmental disturbances; differential encoding of transformed coefficients was shown to be very helpful as well [11]. However, while all these approaches have proven more robust than the conventional SVQ (i.e., performing SVQ on Mel frequency cepstral coefficient (MFCC) directly), they are still based on VQ or SVQ, which are distance- and codebook-based. As long as the quantization is based on a pretrained codebook and some distance measure with the codebook, the mismatch between VQ codebook and testing feature vectors under lower signal-to-noise ratio (SNR) conditions remains a difficult problem.

For the above cases of robust and/or distributed speech recognition, feature vectors corrupted by environmental noise and/or quantization errors can be viewed as random vectors with uncertainty. Uncertainty decoding approaches have been proposed to consider such uncertainty [3], [12]–[15], including handling those produced by environmental noise [12]–[14] and estimating the uncertainty generated in the quantization process [3], [15]. However, in DSR with environmental noise, it is naturally better to consider environmental noise and quantization errors jointly. However, this is difficult because environmental noise is hidden in the quantized codewords, or mixed with quantization errors. The meager computational resources available on hand-held devices further complicate many useful advanced robust approaches. Furthermore, when noise conditions are unknown and/or are changing at the moving client, various successful data-driven robust methods cannot be used. The recommendation to use a standardized VQ codebook also leads to further difficulties because of the inevitable codebook mismatch.

In addition to quantization distortion and environmental noise, in DSR cases the transmission errors caused by communication channels create further problems. Various error concealment (EC) techniques have been proposed to handle these transmission errors. Some reduce transmission errors through error detection and correction [16], some reconstruct the feature vectors by estimating the erroneous subvectors [17], and some consider the reliability of the estimated vectors at the decoding stage [18]–[20]. These methods are very useful

Manuscript received April 26, 2007; revised January 15, 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Abeer Alwan.

The authors are with the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 10617, Taiwan, R.O.C. (e-mail: d94942018@ntu.edu.tw; lslee@gate.sinica.edu.tw).

Digital Object Identifier 10.1109/TASL.2008.920891

when the input speech is clean, in which case it is possible to make up for transmission errors because there are enough correctly received feature parameters, and the continuity nature or prior statistical information of speech signals can be useful in data consistency checks [21] or lost vectors estimation [17]. However, it is important to consider the effectiveness of these methods when the input speech is seriously corrupted by environmental noise.

In this paper, histogram-based quantization (HQ) is proposed to solve the many related problems mentioned above. HQ is a novel approach in which the partition cells for quantization are dynamically defined by the histogram or order statistics of a segment of recent past samples of the parameter to be quantized. It is actually a dynamic quantization, completely based on the local statistics of the signal, not on any distance measure, nor directly related to any pretrained codebook. On one hand, in the case of DSR, many of the above-mentioned problems that arise from a fixed pretrained VQ codebook in conventional DSR framework are shown to be solved to a good extent with this new approach, because the quantization is dynamic and not solely based on a fixed pretrained codebook at all; therefore, the mismatch between the corrupted feature vectors and a fixed pretrained codebook is reduced. This concept of HQ is then further extended to histogram-based vector quantization (HVQ). On the other hand, HQ is also shown to be useful as a good approach for robust feature transformation, which can produce more robust features, because most of the noise disturbances can be automatically absorbed by the dynamic histogram. This robust nature of HQ against environmental noise is extensively explored and analyzed, including considering quantization resolution (or required bit rate), noisy environment, and transmission conditions. The quantization distortion and environmental noise are jointly considered further in a joint uncertainty decoding (JUD) approach for HQ. For robust speech recognition alone without DSR, HQ can be used as the front-end feature transformation and JUD as the enhancement approach at the back-end recognizer. For DSR applications, on the other hand, HQ can be applied at the client end as a quantization process for data compression, and JUD at the server. In addition, a three-stage EC framework is further proposed for a DSR transmission environment to handle transmission errors introduced by wireless channels, in which the first stage detects the erroneous feature parameters, the second stage reconstructs the detected erroneous subvectors, and the third stage considers the uncertainty of the estimated vectors during Viterbi decoding. All the claims mentioned above were verified by extensive experiments reported below that were performed under the AURORA 2 testing environment for different types of noise, different SNR values, and different transmission conditions including different bit rates [22].

The rest of this paper is organized as follows. In Section II, the complete formulation of HQ is presented and its robust nature discussed. Section III then discusses JUD for HQ, and Section IV presents the three-stage EC approach. In Section V, the experimental setup is described. The many results for a whole series of experiments for both robust and/or distributed speech recognition are then presented and analyzed in detail in Section VI. The concluding remarks are finally made in Section VII.

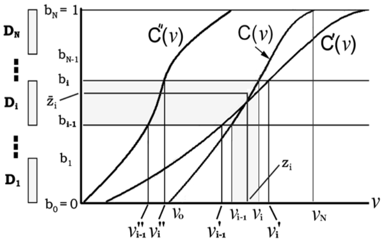

Fig. 1. General formulation of histogram-based quantization (HQ).

II. Histogram-Based Quantization (HQ)

A. General Formulation of HQ

The concept of HQ is to perform quantization of a feature parameter y_t at time t based on the histogram or order statistics of that feature parameter within a moving segment of the most recent past T samples, $[y_{t-T+1}, \ldots, y_{t-1}, y_t] \triangleq Y_{t,T}$, up to the time t being considered [23]. As shown in Fig. 1, the values of these T parameters in $Y_{t,T}$ are sorted to produce a time-varying cumulative distribution function $C(v)$, or histogram, which changes for every time instant t, where $C(v_0) = b_0 = 0$ and $C(v_N) = b_N = 1, v_0$ and v_N are, respectively, the minimum and maximum values within $Y_{t,T}$. Also shown in Fig. 1, N partition cells, $\{D_i = [b_{i-1}, b_i], i = 1, 2, \ldots, N\}$, together with their corresponding representative values, $\{\bar{z}_i, i = 1, 2, \ldots, N\}$, are defined on the vertical scale [0, 1], which are derived from a standard Gaussian $N(0, 1)$ with cumulative distribution $C_0(v)$ via the Lloyd–Max algorithm [24], [25]. Note that the boundaries $\{b_i, i = 0, 1, 2, \ldots, N\}$ on the vertical scale can be either uniformly or nonuniformly distributed [23]. In the case of nonuniform quantization, the Lloyd–Max algorithm can be performed with respect to any distribution, including the distribution of training sets. Since different training sets may have different distributions, we performed the Lloyd–Max algorithm based on uniform, Laplacian and Gaussian distributions in the preliminary experiments. The best performance was obtained with Gaussian distribution under noisy environments, probably because the distribution of feature parameters under noisy environments on the vertical scale is closer to a Gaussian distribution. Using the dynamic histogram $C(v)$ constructed with $Y_{t,T}$, these partition cells on the vertical scale, $\{D_i, i = 1, 2, \ldots, N\}$, are then transformed to the horizontal scale to be the N partition cells $[v_{i-1}, v_i], i = 1, 2, \ldots, N$ on the horizontal scale for the quantization of y_t, where $C(v_i) = b_i$. In other words, the partition cell $[v_{i-1}, v_i]$ on the horizontal scale is obtained from the partition cell $D_i = [b_{i-1}, b_i]$ on the vertical scale via the dynamic histogram $C(v)$. Thus, the partition cell $[v_{i-1}, v_i]$ on the horizontal scale is dynamic. However, the representative values $\{z_i, i = 1, 2, \ldots, N\}$ for these partition cells $\{[v_{i-1}, v_i], i = 1, 2, \ldots, N\}$ on the horizontal scale are fixed, and are transformed from the representative values $\{\bar{z}_i, i = 1, 2, \ldots, N\}$ previously obtained on the vertical scale by the histogram $C_0(v)$ of the standard Gaussian.

The above formulation indicates that HQ is based on a hidden codebook $\{(D_i, \bar{z}_i), i = 1, 2, \ldots, N\}$ derived from a standard Gaussian on the vertical scale, which is then transformed by a dynamic histogram $C(v)$ into time-varying partition cells $[v_{i-1}, v_i]$, and by a fixed histogram $C_0(v)$ into the fixed representative values z_i, both on the horizontal scale. The quantization here is then similar to all conventional quantization processes, in that it is a mapping relation which maps the present parameter y_t to a fixed representative value z_i, if y_t is within the partition cell $[v_{i-1}, v_i]$, except that this partition cell is dynamically defined

$$y_t \rightarrow z_i, \text{ if } b_{i-1} < C(y_t) < b_i, \text{ or } v_{i-1} < y_t < v_i$$
$$C(v_{i-1}) = b_{i-1}, C(v_i) = b_i, \; i = 1, 2, \ldots, N. \quad (1)$$

Note that the quantization codebook here includes a set of dynamic partition cells $\{[v_{i-1}, v_i], i = 1, 2, \ldots, N\}$ and a set of fixed representative values $\{z_i, i = 1, 2, \ldots, N\}$. It will be shown below that many practical problems mentioned previously can be automatically solved to a good extent in this way. Also, although here HQ is a quantization process, it can also be used as a feature transformation process offering the desired robustness as will also be discussed below, in which each parameter y_t is transformed to its representative value z_i for the corresponding partition cell.

B. Histogram-Based Vector Quantization (HVQ)

The above general formulation of one-dimensional HQ in Fig. 1 can be easily extended to HVQ with more than one dimension. Consider SVQ as an example [4], in which two MFCC parameters (e.g., c_1 and c_2) can be quantized jointly by a two-dimensional VQ codebook. Extending from the one-dimensional HQ mentioned above, a moving segment of the most recent past T samples of the first parameter $y_t^{(1)}$ up to time t, $[y_{t-T+1}^{(1)}, \ldots, y_{t-1}^{(1)}, y_t^{(1)}] \triangleq Y_{t,T}^{(1)}$, gives a histogram $C_1(v^{(1)})$ for $y_t^{(1)}$, and a similar segment of the past T samples of the second parameter $y_t^{(2)}$ up to time t, $Y_{t,T}^{(2)}$, gives another histogram $C_2(v^{(2)})$ for $y_t^{(2)}$. The formulation below is exactly the same as the one-dimensional HQ in Fig. 1, except that here both the vertical and horizontal axes are no longer one-dimensional axes, but are extended to vertical and horizontal two-dimensional planes as shown in Fig. 2. On the vertical plane with coordinates $(b^{(1)}, b^{(2)})$, we have a two-dimensional hidden codebook $\{(D_i, \bar{z}_i), i = 1, 2, \ldots, N\}$, which is derived from a bivariate standard Gaussian via the LBG algorithm [26]. Every point $(b^{(1)}, b^{(2)})$ on this plane is then transformed by the above-mentioned dynamic histograms $C_1(v^{(1)}), C_2(v^{(2)})$ back to a point $(v^{(1)}, v^{(2)})$ on the horizontal plane, where $C_1(v^{(1)}) = b^{(1)}, C_2(v^{(2)}) = b^{(2)}$. The set of all these points $(v^{(1)}, v^{(2)})$ on the horizontal plane transformed from those points $(b^{(1)}, b^{(2)})$ on the vertical plane in a certain partition cell D_i then forms the dynamic partition cell Q_i on the horizontal plane as follows:

$$(v^{(1)}, v^{(2)}) \in Q_i, \text{ if } (b^{(1)}, b^{(2)}) \in D_i$$
$$C_1(v^{(1)}) = b^{(1)}, \; C_2(v^{(2)}) = b^{(2)}, \; i = 1, 2, \ldots, N. \quad (2)$$

On the other hand, the representative points $\bar{z}_i$ for each partition cell D_i on the vertical plane are similarly transformed back to the fixed representative points z_i on the horizontal plane, except that the transformation is performed by two fixed histograms $C_0(v^{(1)})$, $C_0(v^{(2)})$, both derived from a one-dimensional standard Gaussian. The quantization here is a mapping relation just as one-dimensional HQ in (1), which maps the present parameter set $(y_t^{(1)}, y_t^{(2)})$ to a representative value z_i for the dynamically defined partition cell Q_i

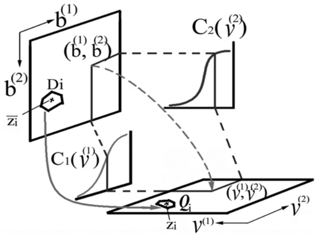

Fig. 2. Concept of histogram-based vector quantization (HVQ) using two dimensions.

$$(y_t^{(1)}, y_t^{(2)}) \rightarrow z_i, \quad \text{if } (C_1(y_t^{(1)}), C_2(y_t^{(2)})) \in D_i$$
$$\text{or } (y_t^{(1)}, y_t^{(2)}) \in Q_i, \; i = 1, 2, \ldots, N. \quad (3)$$

Based on the above, the two-dimensional HVQ can be performed dynamically on the $(v^{(1)}, v^{(2)})$ plane. For the present parameter pair $(y_t^{(1)}, y_t^{(2)})$ at time t, the two dynamic histograms $C_1(v^{(1)})$ and $C_2(v^{(2)})$ based on $Y_{t,T}^{(1)}$ and $Y_{t,T}^{(2)}$ give a point $(C_1(y_t^{(1)}), C_2(y_t^{(2)}))$ on the vertical plane. The partition cell D_i on the vertical plane to which this point belongs then determines the partition cell Q_i and representative point z_i on the horizontal plane.

C. Discussions About Robustness of HQ (and HVQ)

Conventionally, feature quantization is for data compression and robust features are for handling noise disturbances. The proposed HQ, however, includes the desired robustness in the quantization process.

1) Robust Nature of HQ: With the conventional SVQ, the mismatch between the pretrained fixed VQ codebook and the current corrupted testing features may significantly increase quantization distortions. With the proposed HQ, however, the actual partition cells are dynamically adjusted according to local statistics. For example, as shown in Fig. 1, $C(v)$ may be changed to $C'(v)$ when disturbances are encountered. The partition cell on the horizontal scale for the disturbed parameter y'_t may also be changed to $[v'_{i-1}, v'_i]$, where $C'(v'_{i-1}) = b_{i-1}$ and $C'(v'_i) = b_i$, which can be quite different from $[v_{i-1}, v_i]$. Nevertheless, the partition cell D_i and the corresponding representative value z_i for y'_t may remain unchanged as long as $v'_{i-1} < y'_t < v'_i$, since D_i is fixed on the vertical scale, while the disturbances from y_t to y'_t are on the horizontal scale, and z_i is fixed on the horizontal scale. Since the actual partition cells are no longer fixed as in conventional SVQ methods, the codebook mismatch problem mentioned above can thus be avoided to some extent. In other words, HQ is based on the partition cells D_i fixed on the vertical scale and the dynamic histogram $C(v)$, and is therefore less sensitive to disturbances on the horizontal scale: disturbances on the horizontal scale are actually absorbed by the dynamic histogram to a certain degree. When a segment of parameters $Y_{t,T}$ are corrupted by small disturbances, all individual values may be changed ($C(v)$ is disturbed into $C'(v)$), but the order statistics which produce the partition cells on the horizontal scale may remain similar, and the representative values z_i remain fixed; therefore, the changes to the quantization results may be very limited. Such robustness is obtained by local order statistics for the most recent past values of feature parameter. This is why HQ is able to handle various noise conditions as will be shown in the experiments presented below.

2) Comparison With Histogram Equalization (HEQ): The popularly-used HEQ equalizes the cumulative distributions (or histograms) of both the training and testing feature parameters in each temporal span, and has been shown to produce very robust features for recognition [8]–[10]. HQ actually borrows the concept from HEQ. The experiments below will show that HQ can be used as an attractive feature transformation approach for robustness purposes as well, and it even performs better than HEQ. It is important to explain why. HEQ actually performs point-to-point feature transformation based on the order statistics, which can absorb the small disturbances to a good degree, although some residual disturbances inevitably remain because the point-based order statistics are in any case more or less disturbed. Quantile-based HEQ [27] performs a piecewise-linear approximation of HEQ. It reduces the computation complexity for histogram estimation, but does not change the point-based nature of the transformation. HQ, on the other hand, performs the transformation block by block; therefore, the small disturbances within each block (D_i in Fig. 1) are absorbed by the block-based order statistics. The block-based order statistics certainly introduce uncertainty as well, but with the proper choice of the number of quantization levels N or the block size, this uncertainty may be compensated for by the stochastic nature of the Gaussian mixtures in the HMMs. HEQ can be considered the limiting case of HQ when the number of quantization levels N becomes infinite. As will be shown below, the recognition performance certainly depends on the value of N considering the noise conditions and so on, but N being infinite is not necessarily the best.

III. Joint Uncertainty Decoding (JUD) for HQ

Uncertainty decoding has been developed for HMM decoding considering the uncertainty of the observation vectors. Such techniques are also very useful for the HQ developed here, as presented below.

A. General Formulation of Uncertainty Decoding

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

$$b_j(w) = \sum_{m=1}^{M} c_{jm} N(w; \mu_{jm}, \Sigma_{jm}) \tag{4}$$

where m is the mixture index, and $c_{jm}, \mu_{jm}, \Sigma_{jm}$ are, respectively, the mixture weight, mean, and covariance for the mth Gaussian mixture in state j. There have been slightly different approaches in formulating the concept of uncertainty decoding [12], [14]. In the approach used here [3], [13], [15], instead of evaluating the observation probability $b_j(w)$ only for a single feature vector w, uncertainty decoding treats the observed feature vector w as being corrupted, and therefore considers the uncorrupted but unobservable feature vector o as a random variable with a distribution $p(o|w)$ during decoding. The probability of observing w, $b_j(w)$, can then be defined as the expected value of $b_j(o)$ with respect to the distribution $p(o|w)$ [3], [13], [15]

$$b_j(w) = E_{o|w}([b_j(o)]) = \int_o p(o|w) b_j(o) do. \tag{5}$$

Assuming $p(o|w)$ to be Gaussian with mean $\mu_{o|w}$ and covariance matrix $\Sigma_{o|w}, p(o|w) \sim N(o; \mu_{o|w}, \Sigma_{o|w})$, where both $\mu_{o|w}$ and $\Sigma_{o|w}$ can be estimated in various ways, the integration in (5) can be reduced to [13]

$$b_j(w) = \sum_{m=1}^{M} c_{jm} N(\mu_{o|w}; \mu_{jm}, \Sigma_{jm} + \Sigma_{o|w}). \tag{6}$$

Thus, the standard HMM decoding using (4) remains unchanged, except that the variance of each Gaussian in the HMMs is increased by $\Sigma_{o|w}$, the uncertainty of the unobservable vector o. In this way, the Viterbi decoding can be based more on reliable parameters with a smaller variance $\Sigma_{o|w}$. The observed feature vector w can be taken as the estimated value of $\mu_{o|w}$ for simplicity, as is done here in this section. However, $\mu_{o|w}$ can also be estimated based on previous feature vectors as in the three-stage error concealment approaches as discussed later on. Below, we present the approaches used here to estimate the uncertainty of the unobservable feature vector o, or the covariance matrix $\Sigma_{o|w}$.

B. JUD for HQ

There are two sources of uncertainty in HQ-based features: quantization errors and environmental noise. Here, we first separately estimate them and then consider them jointly.

1) Quantization Error Uncertainty: In an HQ partition cell, the representative value z_i is the observed corrupted feature vector w in (5), and all the possible samples in the corresponding ith partition cell $[v_{i-1}, v_i]$ are these samples for the uncorrupted unquantized feature vectors o in (5) collected at the client, which are unobservable at the server. The variance $\Sigma_o^{q,i}$ for quantization errors in the ith partition cell to be used to take the place of $\Sigma_{o|w}$ in (6) can thus be estimated using a clean speech training set. Taking the one-dimensional HQ as in Fig. 1 as an example

$$\Sigma_o^{q,i} = \frac{1}{L_i} \sum_{v_{i-1}<y_t<v_i} \left(C_0^{-1}[C(y_t)] - z_i\right)^2 \tag{7}$$

where the summation is over all L_i feature parameters y_t in the ith partition cell $[v_{i-1}, v_i]$ in the training set. Equation (7) can be easily extended to HVQ for more dimensions. Because the representative value z_i was obtained via the Lloyd–Max algorithm (or LBG algorithm [26] in the case of HVQ) based on the histogram $C_0(\bullet)$ for a standard Gaussian distribution, all parameters y_t in the partition cell need to be transformed first by $C(\bullet)$ then transformed back via $C_0^{-1}(\bullet)$ to evaluate $\Sigma_o^{q,i}$. Because the Lloyd–Max algorithm produces tightly quantized levels in high-density regions and loosely quantized levels in low density regions to minimize total distortion, uncertainty decoding automatically increases the Gaussian variances for the loosely

TABLE I
AVERAGED HISTOGRAM SHIFT FOR HQ UNDER DIFFERENT SNR CONDITIONS

SNR	Clean	20 dB	15 dB	10 dB	5 dB	0 dB
Histogram shift $\lvert C_t^{-1}(0.5)\rvert$	0.016	0.038	0.053	0.090	0.109	0.132

quantized levels. In this way, $\Sigma_o^{q,i}$ can be trained in advance for all partition cells $[v_{i-1}, v_i]$.

2) Environmental Noise Uncertainty: Under low SNR conditions, disturbances may be very serious. For example, in Fig. 1, v_{i-1} and v_i may be changed to v''_{i-1} and v''_i and $C(v)$ to $C''(v)$, or there may be a histogram shift which cannot be well absorbed by the dynamic histogram. Inevitably, then, HQ's performance deteriorates. Such a histogram shift may be reasonably estimated by $C_t^{-1}(0.5)$, because $C_0^{-1}(0.5) = 0$ for a standard zero-mean Gaussian. For server-side histograms constructed based on the quantized codewords, the average values of $|C_t^{-1}(0.5)|$ under all types of noise for the AURORA 2 testing environments (with further details in Section V) for different SNR values are shown in Table I. Clearly, the histogram shift increases with lower SNR values. This is reasonable because under lower SNR conditions, the order statistics and histograms of the original speech samples collected at the client in the respective moving segments change very rapidly; thus, the quantized HQ codewords based on these histograms also change quickly and significantly with time. As a result, the server-side histogram constructed using the quantized HQ codewords also change quickly and significantly with time, introducing a significant and fast fluctuating bias or shift $|C_t^{-1}(0.5)|$ in each short segment, even if the original noise added to the signal samples is zero-mean in the long term. Hence, we can take the histogram shift $|C_t^{-1}(0.5)|$ as a simple indicator for the SNR condition: that is, higher such shifts correspond to lower SNR values. Therefore, the variance $\Sigma_o^{n,t}$ for uncertainty caused by environmental noise at time t—used in place of $\Sigma_{o|w}$ in (6)—can be reasonably estimated as

$$\Sigma_o^{n,t} = \alpha(C_t^{-1}(0.5))^2 \tag{8}$$

where α is an empirically determined scaling factor and is fixed for all SNR values and noise conditions in our experiments. In fact, the value of $\Sigma_o^{n,t}$ only indicates the relative importance of feature parameters in Viterbi decoding—we found in preliminary experiments that recognition performance is not very sensitive to the value of α chosen here. $C_t(\bullet)$ is the histogram for the HQ-quantized codewords z_i for all feature parameters y_t in the moving segment $Y_{t,T}$ at frame t. In this way, in the DSR case, $\Sigma_o^{n,t}$ can be estimated at the server easily for each time t without any extra bit rate costs. This allows us to solve the problem where the environmental disturbances are hidden in codewords and cannot be estimated directly.

3) Joint Uncertainty Decoding (JUD) for HQ: The above two types of uncertainties should be jointly considered [28]. A reasonable assumption is that for higher SNR conditions the quantization error uncertainty $\Sigma_o^{q,i}$ dominates, while for lower SNR conditions, the environmental noise uncertainty $\Sigma_o^{n,t}$ dominates. Therefore, the joint uncertainty $\Sigma_o^{i,t}$ for a codeword z_i in the ith partition cell at time t can be estimated as

$$\Sigma_o^{i,t} = \max(\Sigma_o^{q,i}, \Sigma_o^{n,t}) \tag{9}$$

where $\Sigma_o^{q,i}$ is pretrained for the ith partition cell using (7), and $\Sigma_o^{n,t}$ is estimated in real time using (8). This value of $\Sigma_o^{i,t}$ can then be used as $\Sigma_{o|w}$ directly in (6).

C. Histogram-Shift Compensation

As mentioned previously, histogram shift occurring at lower SNR values inevitably results in seriously degraded HQ performance. As a result, in addition to the uncertainty decoding as mentioned above, we can also shift the histogram horizontally to have

$$C_t^{-1}(0.5) = 0 \tag{10}$$

for each time t. A large portion of the serious disturbances can be absorbed by such a shift, as will be verified by the experiments below.

IV. THREE-STAGE EC FOR HQ-BASED DSR SYSTEMS

Here, we consider the approaches to handling the transmission errors added to the received HQ codewords under the DSR framework [29]. A three-stage EC approach is developed, as presented below.

A. Stage 1—Error Detection

In the ETSI DSR standards, every two frames are grouped together and protected with four-bit CRC [4]. In this way, the entire frame-pair is labeled erroneous even if only a single bit error occurs in the frame-pair packet. Adding check bits at the subvector level is helpful for subvector level error detection, but comes at the cost of additional bandwidth [7]. A more efficient way is to make use of the speech signal characteristics at the subvector level. The data consistency test checks the continuity of the parameters in two neighboring subvectors [21]. When the difference between two consecutive values of a feature parameter in a subvector exceeds a predetermined threshold obtained from some training corpus, the subvector is classified as inconsistent. However, if the statistics of the testing features are time-varying and different from those of the training corpus, this approach becomes less reliable. With environmental noise, the parameters are likely to be classified as inconsistent even if they are correctly received.

HQ performs feature parameter quantization based on the local histogram (or order statistics), so the quantized codewords represent the local order-statistic information of the original parameters. The quantization process does not change the order statistics of the parameters, and if there are no transmission errors, the histogram for the subvector codewords received at the server should be similar to the histogram for the original feature parameters at the client. Thus, the partition cell obtained by reperforming HQ on the received subvector codeword, based on the dynamic histogram for these received codewords, should

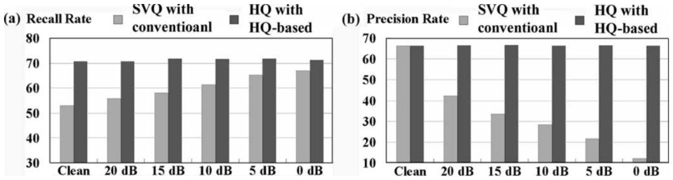

Fig. 3. (a) Recall and (b) Precision rates for error detection using SVQ with the conventional data consistency check and HQ with the HQ-based consistency check proposed here.

be the original partition cell. If not, it is very possible that the order statistics have been changed and the received subvector codeword may be erroneous. Based on this observation, the consistency test in the HQ framework proposed here is as follows. Taking a two-dimensional HVQ as an example, $z_i = (z_i^{(1)}, z_i^{(2)})$ is a received subvector codeword at some time, and $\mathrm{HQ}\{(z_i^{(1)}, z_i^{(2)})\}$ represents the representative value for the subvector $(z_i^{(1)}, z_i^{(2)})$ assigned by HQ performed at the server based on the histogram for the received codewords. The subvector $(z_i^{(1)}, z_i^{(2)})$ is then classified as consistent if

$$HQ\{(z_i^{(1)}, z_i^{(2)})\} = (z_i^{(1)}, z_i^{(2)}). \tag{11}$$

In other words, if these two parameters are correctly received, their order statistics at the server should be similar to the order statistics for the original values before quantization at the client, and therefore similarly quantized into the same HQ partition cell.

We compared the error detection accuracy of the conventional SVQ scheme with the data consistency check [21] and the proposed HQ with the HQ-based consistency check mentioned above under all different noise conditions for the AURORA 2 testing environment with the transmission errors introduced by the General Packet Radio Service (GPRS) wireless environment (further details are presented in Section V below). The averaged recall (percentage of detected errors out of all errors) and precision (percentage of correct errors out of all detected errors) rates for error detection are shown in Fig. 3(a) and (b). For lower SNR cases, it is clear that the noise seriously affects the SVQ with data consistency check as verified by the precision degradation in Fig. 3(b) (from 66% at clean down to 12% at 0 dB). With the proposed HQ-based consistency check approach, however, the precision rate is much more stable at all SNR values, and both recall and precision rates are higher.

Note that when (11) is not satisfied, it is also possible that the present codeword is actually correctly received, but instead the dynamic histogram, on which the HQ in (11) is based, is disturbed by erroneous received codewords in the past T frames. This is one good reason why the precision rate in Fig. 3(b) for HQ with the proposed consistency check is slightly less than 70%, i.e., some detected inconsistencies are actually correctly received codewords. However, this precision is much higher than SVQ with conventional approach. In fact, the probability that the inconsistency in (11) is due to the disturbed histogram rather than the considered codeword being erroneous is lower, because the effect of the erroneous codewords in the past T frames is reasonably absorbed by the histogram (the order statistics of a large number of codewords) as well as the partition cells in HQ. In other words, with erroneous codewords in the past T frames, the change of the histogram may not be very serious, and the partition cell that the present codeword being considered belongs to may remain unchanged. This is verified in Fig. 3(b) where the precision rate, although much less than 100%, remains almost the same from clean speech to 0-dB SNR.

B. Stage 2—Erroneous Feature Vector Estimation

Different techniques for estimating the detected erroneous feature vectors have been proposed. Repetition and interpolation only use the correctly received feature vectors [16], while statistical-based techniques use prior knowledge about speech source in addition, and have been shown to offer better performance [30].

The erroneous subvector estimation proposed here under the HQ framework is based on the maximum *a posteriori* (MAP) criterion, which determines the estimated value $\hat{s}_t$ of a certain transmitted subvector codeword s_t at time t, which is detected as erroneous (here both $\hat{s}_t$ and s_t are certain codewords z_i mentioned above for some i, respectively). This MAP estimation is conditioned on the present and previously received corresponding subvector codewords r_t and r_{t-1} (here both r_t and r_{t-1} are also certain codewords z_i mentioned above for some i, respectively)

$$\hat{s}_t = \arg\max_{z_i}\{P(s_t = z_i | r_t, r_{t-1})\} \tag{12}$$

where $s_t = z_i$ denotes that s_t is the ith HQ codeword out of the N possible codewords. The maximization here is over all of these codewords. If we assume r_t and r_{t-1} are independent

$$P(s_t | r_t, r_{t-1}) \approx \frac{P(s_t | r_{t-1}) P(s_t | r_t)}{P(s_t)} = \frac{P(s_t | r_{t-1}) P(r_t | s_t)}{P(r_t)}. \tag{13}$$

With the denominator in (13) left out in the maximization in (12), the probability in (12) can be approximated by the codeword bigram $P(s_t = z_i | r_{t-1})$ and the channel transition probability $P(r_t | s_t = z_i)$

$$\hat{s}_t = \arg\max_{z_i}\{P(s_t = z_i | r_{t-1}) P(r_t | s_t = z_i)\}. \tag{14}$$

In (14), the codeword bigram $P(s_t = z_i | r_{t-1})$ can be estimated by the bigram of the considered subvector codewords $P(s_t = z_i | s_{t-1})$ trained from a clean training set (for example, the clean training set of AURORA 2). Also, the channel transition probability $P(r_t | s_t = z_i)$ in (14) can be estimated from the bit error rate (BER) of the present frame being considered

$$P(r_t | s_t = z_i) = BER^{d[b(z_i), b(r_t)]} * (1 - BER)^{K - d[b(z_i), b(r_t)]} \tag{15}$$

where BER is estimated as the total number of inconsistent subvectors (in simulation analysis, it was found that in most cases there is only one bit error in an erroneous codeword, and therefore this number can be used to estimate the total number of erroneous bits) detected in the first stage (discussed in Section IV-A) in the present frame divided by the total number of bits in the frame, K is the total number of bits in the received subvector codeword r_t, $b(z_i)$ and $b(r_t)$ are, respectively, the bit patterns for the codewords z_i and r_t, and $d(\bullet, \bullet)$ represents the Hamming distance between two bit patterns.

TABLE II
MUTUAL INFORMATION $I(s_t, s_{t-1})$ FOR SVQ AND HQ

$I(s_t, s_{t-1})$	c_1, c_2	c_3, c_4	c_5, c_6	c_7, c_8	c_9, c_{10}	c_{11}, c_{12}	$c_0, logE$
SVQ	1.365	0.998	0.791	0.652	0.611	0.568	1.455
HQ	1.473	1.110	0.856	0.722	0.678	0.619	1.541

The value of $P(r_t|s_t = z_i)$ in (15) is actually the probability of z_i being changed to r_t if BER can be accurately estimated. With (15), when r_t is less reliable (or has a larger BER), the values of $P(r_t|s_t = z_i)$ for all possible codewords z_i with different i become closer to each other (i.e., the difference in $P(r_t|s_t = z_i)$ is insignificant for different Hamming distances $d(\bullet, \bullet)$). On the other hand, when r_t is more reliable (or has a smaller BER), $P(r_t|s_t = z_i)$ is larger for only few values of i. In this way, more emphasis can be put on the codeword bigram $P(s_t = z_i|r_{t-1})$ than on the channel transition probability $P(r_t|s_t = z_i)$ in (14) when the channel condition is less reliable.

Because the basic principle here is to exploit the short-time correlation between consecutive frames in speech signals to estimate the lost subvectors, the robustness of HQ as mentioned in Section II-C is very helpful. If the quantization process is less robust, the environmental noise may move the feature vectors to a different partition cell and the subvector transition relationship in speech signals may be disturbed. This problem is actually lessened by the HQ's robustness, as can be verified by the mutual information $I(s_t, s_{t-1})$ between the present and previous subvector codewords s_t and s_{t-1}

$$I(s_t, s_{t-1}) = H(s_t) - H(s_t|s_{t-1}) \tag{16}$$

where

$$H(s_t) = \sum_{j=1}^{N} -P(s_t = z_j) \log[P(s_t = z_j)] \tag{17}$$

and

$$H(s_t|s_{t-1}) = \sum_{i=1}^{N}\sum_{j=1}^{N} -P(s_t = z_j, s_{t-1} = z_i) \log[P(s_t = z_j|s_{t-1} = z_i)] \tag{18}$$

are, respectively, the degree of uncertainty for the present subvector s_t, and the remaining degree of uncertainty for s_t after the previous subvector s_{t-1} is known. Thus, the mutual information $I(s_t, s_{t-1})$ in (16) shows how much the codeword bigram model reduces uncertainty for the subvectors s_t. In other words, a bigram model with higher mutual information implies that predicting the present subvector s_t given the previous subvector s_{t-1} is easier. The mutual information for the conventional SVQ and the proposed HQ averaged for different subvectors from the three testing sets of AURORA 2 is listed in Table II. We can see that HQ's mutual information is always higher than that of SVQ, which indicates that the HQ framework allows for more precise estimation of the lost subvectors.

C. Stage 3—Uncertainty Decoding

The uncertainty decoding discussed in Section III-A can be used here in the final stage. Consider Section III-A: the above received codeword r_t is taken as the observed corrupted feature vector w in (5), and all of the possible transmitted codewords, $s_t = z_i$, $i = 1, 2, \ldots, N$, are the possible samples of the uncorrupted but unobservable feature vector o in (5). The distribution of the probability $P(s_t = z_i|r_t, r_{t-1})$ obtained in (12) then characterizes the uncertainty of the observed codeword. With the estimated codeword $\hat{s}_t$ in (12) taken as the mean $\mu_{o|w}$ and the covariance estimated using the probability distribution $P(s_t = z_i|r_t, r_{t-1})$ taken as the covariance $\Sigma_{o|w}$, both used in (6), uncertainty decoding can then be directly performed within the HQ framework as presented previously by increasing the variance of each Gaussian mixture by $\Sigma_{o|w}$ in the HMMs as in (6) [28]. In this way, HMM decoding puts more emphasis on more reliable subvectors, i.e., those with lower covariance $\Sigma_{o|w}$ for the probability distribution $P(s_t = z_i|r_t, r_{t-1})$ in (12).

D. Three-Stage EC Under the HQ Framework

The three stages of EC under the HQ framework can be easily integrated. At the first stage, the received frame-pairs are first checked with CRC to detect errors at the frame level. The erroneous frame-pairs are then further checked at the subvector level by the HQ consistency test as mentioned in Section IV-A. At the second stage, the erroneous subvectors detected at the first stage are estimated and reconstructed as presented in Section IV-B. At the third stage, uncertainty decoding in the Viterbi search process makes the HMMs less discriminative for subvectors with higher uncertainty as presented in Section IV-C.

V. EXPERIMENTAL CONDITIONS

All the experiments reported in this paper were conducted on the AURORA 2 testing environment [22] based on a corpus of English connected digit strings. Two training conditions (clean-condition and multicondition) and three testing sets (sets A, B, and C) were defined in AURORA 2. Both clean and noisy speech signals were prepared by filtering the TI database (both training and testing) using a telephone-bandwidth bandpass filter. The testing set A included four types of noise which were used in the multicondition training (subway, babble, car, and exhibition), while the testing set B included another four types of noise not used in the multicondition training (restaurant, street, airport, and train station). The testing set C was filtered with a MIRS (Modified Intermediate Reference System, which simulates the bandpass filtering [300–3400 Hz] behavior of the telephone channels in the public switched telephone networks [PSTN]) characteristic filter [22], [31] before adding two additive noise types (subway in set A and street in set B). In all sets A, B, and C, the SNR tested ranged from 20 to −5 dB. The MFCC extraction follows the WI007 front-end [22] defined in AURORA 2 with frame length 25 ms and frame shift 10 ms, which gives 13 coefficients (C1-C12 and log energy) to be used to obtain the delta and delta-delta features together for recognition.

General Packet Radio Service (GPRS) was chosen in this research as an example for wireless channels in the experiments;

GPRS was developed by ETSI based on a packet switching framework to enhance the GSM system. GPRS shares the GSM frequency bands and uses several properties of the physical layer of the GSM system. It includes four different error control coding schemes, CS1-CS4, each with a different code rate. The GPRS simulation software used in the tests described here was developed by the Wireless Communication Laboratory of National Taiwan University [32], in which all complicated transmission phenomena have been carefully simulated in detail, such as the propagation model, multipath fading, Doppler spread, etc. The experimental results presented below are based on the following simulation configurations: typical urban (TU, an environment more frequently encountered with a more severe fading problem), the client traveling at speeds of 3, 50, 100, 250 km/h, single antenna, hard decision at the receiver, and CS4 (i.e., without any protection) coding scheme, which corresponds to a transmission bit error rate of 5.3% for a client traveling at a speed of 3 km/h.

VI. Experimental Results

The fundamental experimental results for HQ as discussed in Sections II-A–C are briefly reported in sections Sections VI-A–C. Sections VI-D and VI-E then present the results for robust and distributed speech recognition systems, respectively. All the experiments reported here were based on order statistics over segments of most recent past parameter values as mentioned in Section II, so there was no time delay. Better results were obtainable if this no-delay condition was removed.

A. HQ as a Feature Transformation Method

In the first set of experiments, we considered the case of robust speech recognition apart from the DSR environment, in which one-dimensional HQ was used as a feature transformation technique, that is, each feature parameter y_t is transformed to the representative value z_i for the corresponding partition cell as in (1) to be used for recognition.

The results are shown in Fig. 4(a)–(c). The recognition accuracies for baseline experiments with original MFCC features, compared to those with MFCC parameters filtered by the MVA filter (mean and variance normalization followed by autoregression moving-average (ARMA) filtering) [33] and the principal component analysis (PCA) filter derived [34], as well as transformed by the well-accepted HEQ [8]–[10], and the proposed one-dimensional HQ are, respectively, shown in Fig. 4 under clean-condition training for (a) averaged over all SNR values but separated for different types of noise, (b) averaged over all types of noise but separated for different SNR values, and (c) averaged over all types of noise and all SNR values for testing sets A, B, and C, respectively. Here, the order of the MVA filter was $M = 2$, the PCA filter was performed with filter length $L = 15$, and HEQ was performed in exactly the same way as HQ, based on a moving segment of the most recent T past parameters, and the same value of $T = 100$ (or one second) was used for all experiments for both HEQ and HQ. It has been verified that long-term features derived from one second time interval carry important speech information [35].

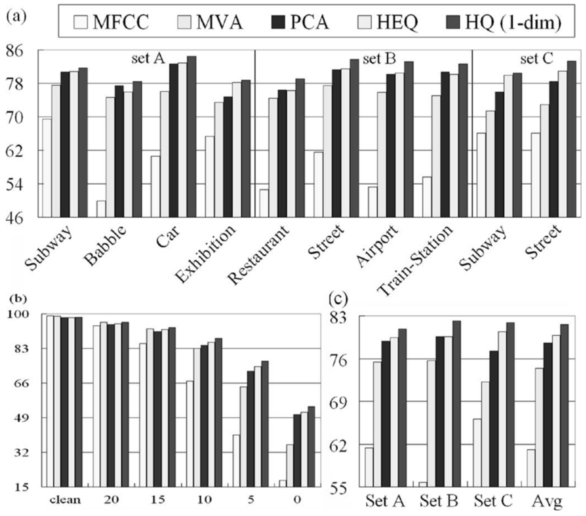

Fig. 4. Accuracies for MFCC baseline and those transformed by MVA filtering, PCA filtering, HEQ, and HQ, respectively, under clean condition training. (a) Averaged over all SNR values but separated for different types of noise. (b) Averaged over all types of noise but separated for different SNR values. (c) Averaged over all types of noise and all SNR values for different testing sets.

Many observations can be made here. First, it is clear that HQ (the last bar) significantly improved the performance as compared to the baseline MFCC (the first bar) for all testing sets, all SNR values (except for the clean speech case), and all noise types. For example, from Fig. 4(a), it can be observed that for speech-like noise such as babble or restaurant noise, the MFCC baseline accuracy (around 50%) was much lower as compared to most other noise types (around 60% or more). HQ was able to absorb the speech-like variation and improved the performance in such a way that the results for different noise types were not only much higher, but also were more similar to each other (around 80%). As another example, in Fig. 4(b) the recognition accuracy of HQ was 87.88% as compared to MFCC baseline 66.95% at 10-dB SNR. The improvements became even more significant for lower SNRs. Second, HQ proposed here performed consistently better than MVA, PCA, and HEQ compared here for all testing sets, all noise types, and all SNR conditions (except for clean speech cases). In particular, HEQ and HQ (the fourth and fifth bars) performed better as compared to MVA and PCA (the second and third bars). This is probably because HEQ and HQ dynamically transform the MFCC features considering the whole distribution locally, while the filters used in MVA and PCA are fixed, and only the first and second moment statistics are taken into consideration. Furthermore, in all Fig. 4(a)–(c), HQ performed consistently better than HEQ for all testing sets, all noise types, and all SNR conditions. For example, in Fig. 4(a), HQ turned out to be very helpful for babble/restaurant noise (78.41%/79.08%) as compared to HEQ (75.95%/76.28%), probably because in such cases of speech-like noise, the order statistics disturbances were better absorbed by HQ's blocks than by HEQ's point-by-point transformation. For subway noise, on the other hand, the improvement of HQ (81.70%) compared to HEQ (80.86%) is relatively less, probably because the impulse-like disturbances may very often exceed beyond the blocks.

TABLE III
AVERAGED NORMALIZED DISTANCES BETWEEN CLEAN AND CORRUPTED SPEECH FEATURES UNDER DIFFERENT SNR VALUES FOR HEQ AND HQ (1-D)

SNR	20 dB	15 dB	10 dB	5 dB	0 dB	-5 dB
HEQ	0.7876	0.8695	0.9516	1.0384	1.1314	1.2276
HQ (1-dim)	0.7172	0.7870	0.8588	0.9362	1.0204	1.1087

We further compared HEQ with HQ (one-dimensional) tested here using a different metric, the averaged normalized distance between the corrupted feature parameters $\overline{x}_t$ and the corresponding clean speech feature parameters x_t

$$d = \frac{1}{\sigma T_N} \sum_{t=1}^{T_N} |\overline{x}_t - x_t|, \tag{19}$$

where the average in (19) is performed over all feature parameters in all the testing speech in sets A, B, C, T_N is the total number of frames, and σ is the standard deviation for all the clean feature parameters x_t. Both $\overline{x}_t$ and x_t have been processed by either HEQ or HQ, so the difference $(\overline{x}_t - x_t)$ indicates how the mismatch caused by noise disturbance is reduced by either HEQ or HQ for each individual feature parameter. Smaller values of d imply that the features are less influenced by disturbances, although d is not necessary directly related to recognition accuracy. The results are listed in Table III for different SNR values. We find in the table that the values of d consistently increase as the SNR value degrades, which makes very good sense, and HQ clearly gives smaller values of d in all cases. This may explain from a different perspective why HQ performed better than HEQ.

B. HQ as a Feature Quantization Method

The next set of experiments considered HQ as a feature quantization method in a DSR framework. However, here we first examined the effect of quantization and compression on recognition accuracy, so we assume that the environmental noise was present with the input speech, but there were no transmission errors. For comparison, recognition accuracies for MFCC features with quantization and compression using the standard SVQ [4], the well-known transform coding [5], [7] (i.e., performing quantization in the transformed domain) followed by SVQ (TC-SVQ), the cascade of the HEQ front-end with SVQ (HEQ-SVQ), and the proposed HQ (actually two-dimensional HVQ) for bit rates 4.4, 3.9, 3.3, and 2.7 kb/s are listed, respectively, in Table IV for clean-condition training, averaged over all ten types of noise and all SNR values in sets A, B, and C. The recognition accuracies for baseline experiments with original MFCC features without quantization is 61.08%. Because all these results are averages over all SNR values from 20 down to 0 dB, the numbers here are not very high. Note that the performance of HQ was consistently and significantly better than SVQ, TC-SVQ, and HEQ-SVQ under all transmission bit rates. For example, at bit rate of 2.7 kb/s, the overall accuracy of HQ (82.08%) represented relative error rate reductions of 26.93%, 62.62%, and 64.57%, respectively, as compared to those with HEQ-SVQ (75.47%), TC-SVQ (52.06%), and SVQ (49.43%). It is even significantly higher (with an error rate reduction of 53.96%) than the original unquantized MFCC (61.08%). This was clearly due to the robust nature of HQ, as discussed previously. Note that the original uncompressed MFCC degraded seriously under noisy conditions, but HQ held up quite well. Also note that the performance of SVQ, TC-SVQ, and HEQ-SVQ all degraded significantly under lower bit rates, while the performance of HQ remained very stable for different bit rates, or the performance of HQ is actually relatively insensitive to the quantization resolution N in (1). These results indicate that, with the conventional distance-based quantization (SVQ), even with the more robust feature transformation front-end (TC or HEQ), the quantization distortion and environmental noise still jointly degraded the performance seriously. The HQ approaches, however, were able to reconstruct the feature parameters based on the order statistics or histogram, which automatically absorbed many of the disturbances, therefore offering a much better recognition accuracy.

TABLE IV
RECOGNITION ACCURACIES FOR FEATURE QUANTIZATION AND COMPRESSION WITH CLEAN-CONDITION TRAINING, AVERAGED OVER ALL SNR VALUES AND NOISE TYPES IN SETS A, B, AND C FOR DIFFERENT BIT RATES (4.4 TO 2.7 kb/s)

Bit rates (kb/s)	4.4	3.9	3.3	2.7
unquantized MFCC	61.08			
SVQ	56.51	55.74	51.13	49.43
TC-SVQ	63.41	62.53	60.33	52.06
HEQ-SVQ	79.79	78.89	78.35	75.47
HQ	81.87	81.95	81.74	82.08

The results in Table IV are averaged over all SNR values and all noise types in sets A, B, and C. Further, we see in Fig. 5(a1)–(a4) the detailed accuracies obtained in exactly the same experiments, but separated for different noise types and averaged over all SNR values for different bit rates (4.4, 3.9, 3.3, and 2.7 kb/s), respectively. From Fig. 5(a1)–(a4), we can find that HQ (the last bar in each set) consistently performed much better than the other approaches compared in Table IV (the first four bars in each set). HQ can even handle nonstationary disturbances as well to a good degree, clearly because it is based on the dynamic histogram of the most recent past values. For example, in the case of 3.3 kb/s in Fig. 5(a3), HQ is actually significantly better than HEQ-SVQ (78.82% versus 73.69%, 79.40% versus 73.77%, 83.80% versus 79.37%, and 83.12% versus 77.82% for babble, restaurant, airport, and train-station noise cases, respectively), and the corresponding numbers for MFCC, SVQ, and TC-SVQ approaches were much lower.

C. Further Analysis of Bit Rates Versus SNRs for HQ as a Feature Quantization Method

To see how quantization distortion (or bit rate) mixed with the environmental noise (SNR) in the input speech jointly influences the recognition performance of a DSR system (assuming no transmission errors), the respective accuracies for the same experiments mentioned in Section VI-B and listed in Table IV are further analyzed, respectively, for different bit rates and different SNRs as shown in Fig. 5(b1)–(b6) for clean to 0-dB SNR. For clean speech, SVQ performed the best (although slightly lower than unquantized MFCC) under higher bit rates (4.4, 3.9, and 3.3 kb/s), while for other approaches (TC-SVQ, HEQ-SVQ, and HQ) feature transformation more or

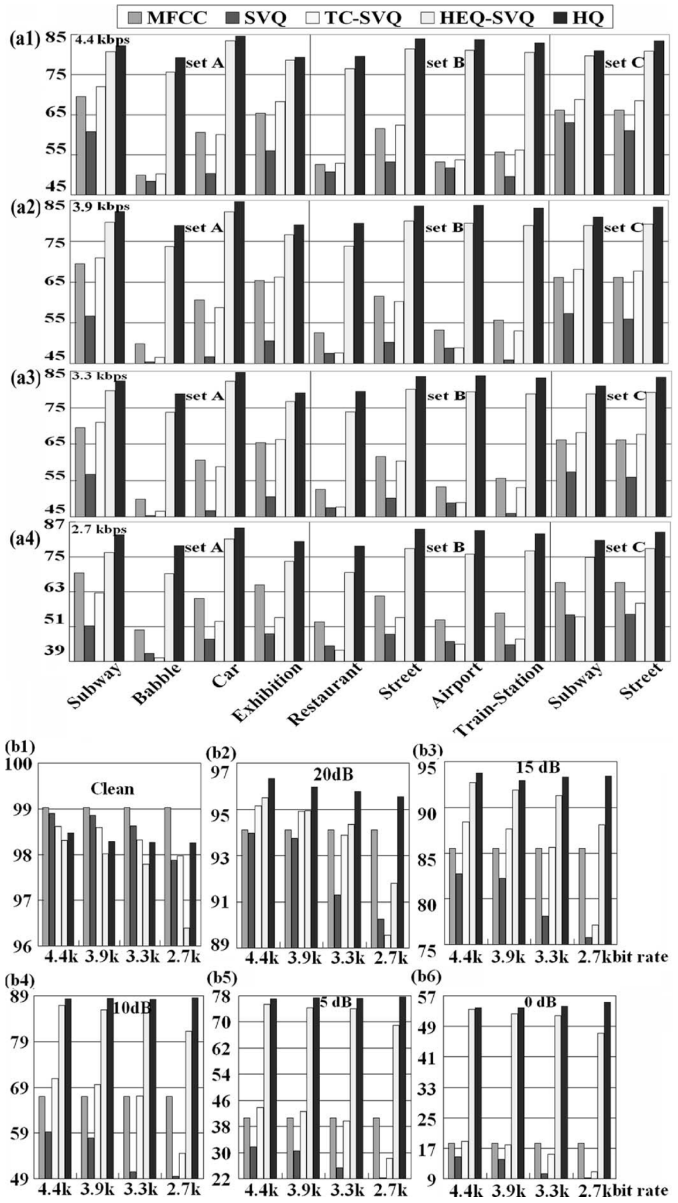

Fig. 5. Recognition accuracies for feature quantization and compression with clean-condition training. (a1)-(a4) Averaged over all SNR values but separated for different types of noise at bit rates of 4.4 to 2.7 kb/s. (b1)-(b6) Averaged over all types of noise but separated for different bit rates (4.4 to 2.7 kb/s) at different SNR values.

less changed the speech characteristics, and therefore inevitably slightly degraded the performance for clean speech. At a lower bit rate such as 2.7 kb/s, however, HQ offered better performance than other approaches. This is probably because SVQ is more sensitive to quantization distortion, so the performance of SVQ, TC-SVQ, and HEQ-SVQ all degraded for lower bit rates. On the other hand, the dynamic nature of HQ makes it relatively insensitive to the quantization resolution (or bit rates), as can be verified in the clean speech case in Fig. 5(b1). Under noisy environments (SNR from 20 dB all the way down to 0 dB), HQ consistently performed better than other approaches for all SNR values and all bit rates. Under very poor SNR conditions, the noisy disturbances were very serious, but still well absorbed by the HQ histogram. For example, in the case of 5-dB SNR and 2.7 kb/s bit rate, HQ offered an accuracy of 77.61% compared to 22.30% for SVQ, 28.31% for TC-SVQ, and 69.07% for HEQ-SVQ. HQ offered an accuracy of higher than 50% (55.27%) even at 0-dB SNR and the low bit rate of 2.7 kb/s. These results indicate that for SVQ the mismatched codebooks significantly increase the quantization distortion, especially under poorer SNR conditions. The performance of HQ, however, remains relatively high and even very stable for different bit rates for SNR degrading from 20 to 0 dB. This verified that HQ is very robust against both quantization distortion and environmental noise.

D. HQ-Based Robust Speech Recognition System With Joint Uncertainty Decoding (JUD)

Here, we consider a complete HQ-based robust speech recognition system under noisy conditions, outside of the DSR or client-server framework. The input speech features were first transformed by HQ just as was presented in Section VI-A. In addition, in this section JUD as discussed in Sections III-A–III-C was further applied at the decoder, including the histogram shift plus the uncertainty estimated for the environmental noise and quantization errors.

The results are plotted in Fig. 6. Note that in Fig. 6(b) the plots for 5- and 0-dB SNR are shown in different scales so as to make the differences easier to observe. The four bars in each set in Fig. 6(a)–(c) are, respectively, for the accuracies obtained with the proposed HQ feature transformation alone (one-dimensional with bit rate (resolution) 3.9 kb/s, exactly the same as the last bar in Fig. 4 presented in Section VI-A), HQ plus histogram shift (HQ-s, Section III-C), HQ with histogram shift plus uncertainty for environmental noise (HQ-s,n, Sections III-C and III-B2), and HQ with complete JUD including histogram shift and uncertainty for environmental noise and quantization errors (HQ-s,n,q, Sections III-C and III-B). It can be found in Fig. 6(a)–(c) that with the various JUD approaches proposed in Sections III-B and III-C performed at the decoder, accuracies can be consistently improved step-by-step in all cases. There was almost no performance degradation for clean speech, and slight improvements at high SNR conditions [Fig. 6(b)]: this implies uncertainty decoding for HQ is able to preserve the discrimination among HMMs. In other words, it is clear that the quantization process 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 to a good extent. Accuracies for the first and the last bars in Fig. 6(c) (HQ alone and HQ-s,n,q with complete JUD) are also compared in Table V. It can be found that significant error rate reduction was actually achieved in all three testing sets.

E. HQ-Based Distributed Speech Recognition (DSR) System

Here, we finally consider a complete DSR system based on the proposed HQ approaches. HQ was first applied at the client end to quantize and compress the input speech features. The quantized codewords were then transmitted via wireless networks to the server. JUD discussed in Section III was then applied at the server to improve accuracies. There were inevitable

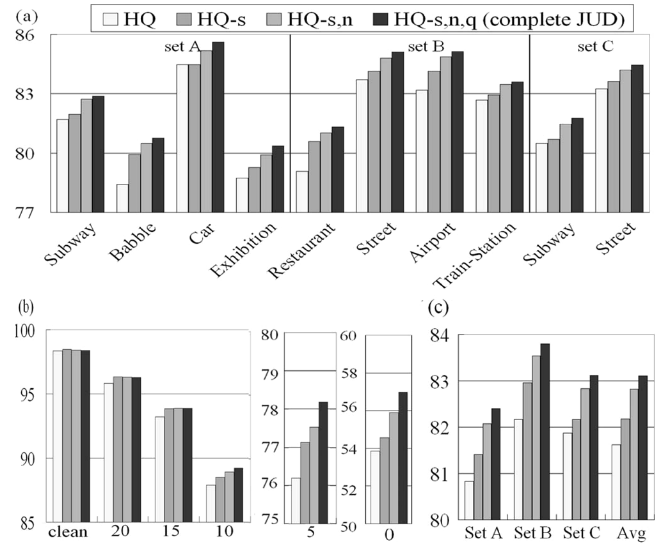

Fig. 6. Performance improvements obtained by the various JUD approaches as compared to HQ alone: (a) averaged over all SNR values but separated for different noise types in sets A, B, and C. (b) Averaged over all noise types but separated for each SNR value. (c) Averaged over all SNR values and noise types but separated into sets A, B, and C.

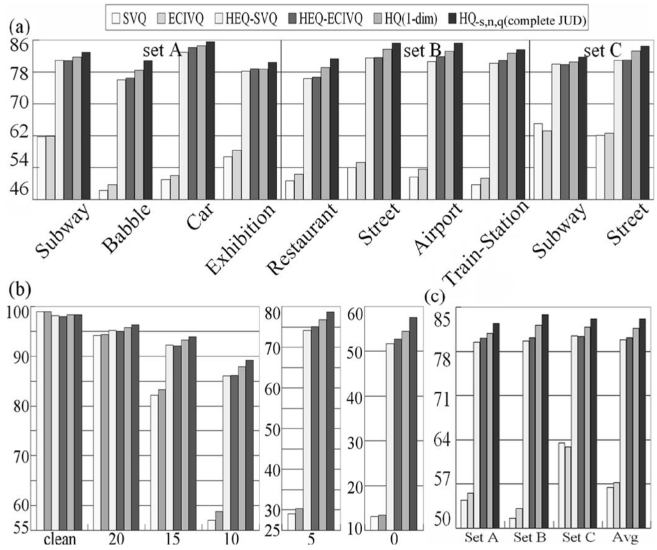

Fig. 7. Comparison of different approaches discussed in this paper for DSR. (a) Averaged over all SNR values but separated for different noise types in sets A, B, and C. (b) Averaged over all noise types but separated for different SNR values. (c) Averaged over all SNR values and noise types but separated for sets A, B, and C.

TABLE V
ACCURACIES AND ERROR RATE REDUCTIONS FOR HQ ALONE (ONE-DIMENSIONAL, 3.9 kb/s) AND HQ-s,n,q (WITH COMPLETE JUD) FOR DIFFERENT TESTING SETS IN FIG. 6(c)

Accuracy	Set A	Set B	Set C	Overall
HQ (one-dimensional)	80.85	82.17	81.86	81.58
HQ-s,n,q (Complete JUD)	82.40	83.81	83.11	83.67
Relative error reduction (%)	8.09	9.14	6.89	8.27

TABLE VI
ACCURACIES AND ERROR RATE REDUCTIONS FOR HEQ-ECIVQ AND HQ-s,n,q (WITH COMPLETE JUD) AT 4.4 kb/s FOR DIFFERENT SNR VALUES IN FIG. 7(b)

SNR	Clean	20 dB	15 dB	10 dB	5 dB	0 dB
HEQ-ECIVQ	98.19	95.25	92.65	86.01	75.96	53.28
HQ-s,n,q(Complete JUD)	98.50	96.38	93.99	89.04	78.34	57.01
Relative error reduction(%)	17.13	23.79	18.23	21.66	9.90	7.98

transmission errors introduced by the wireless channels, and the three-stage EC discussed in Section IV was finally applied.

1) HQ-JUD Compared With Conventional Approaches Associated With SVQ, But Without Transmission Errors: Before considering transmission errors, the first issue to be investigated here is feature quantization and compression. Conventionally, in DSR this is done using SVQ [4]. If noise can be properly handled to a good degree by cascading an HEQ process at the front, we can also compensate for quantization errors caused by SVQ using some conventional approaches associated with SVQ, for example the well-known extended cluster information vector quantization (ECIVQ) [3]. Therefore, we need to compare the proposed HQ followed by JUD with such conventional approaches associated with SVQ first. The results are in Fig. 7(a)–(c). The six bars in each set in Fig. 7 are, respectively, for SVQ alone, ECIVQ alone, the cascade of HEQ front-end and SVQ (HEQ-SVQ), the cascade of HEQ front-end and ECIVQ (HEQ-ECIVQ), HQ (two-dimensional), and the same HQ with complete JUD including histogram shift (HQ-s,n,q), all with bit rates 4.4 kb/s. The first, third, and fifth bars in Fig. 7 are the same as the second, fourth, and fifth bars of the first 4.4-kb/s group in Fig. 5.

We can find from Fig. 7 that ECIVQ (second bar) performed better than SVQ (first bar) for sets A and B, but slightly worse for set C, and the same trend can be observed when HEQ is performed as a front-end of SVQ (HEQ-SVQ, third bar versus HEQ-ECIVQ, fourth bar). This is probably because ECIVQ considers quantization errors only, but the channel mismatch for set C might move the feature vectors to different partition cells, for which the cluster variance used in ECIVQ was not able to help. HEQ offered very significant improvements when cascaded with SVQ or ECIVQ (HEQ-SVQ or HEQ-ECIVQ, third or fourth bar), but the HQ (fifth bar) proposed here consistently provided better performance in almost all cases, and the complete JUD proposed here including histogram shift (HQ-s,n,q, sixth bar) offered additional improvements consistently in almost all cases. The accuracies for HEQ cascaded with ECIVQ (HEQ-ECIVQ, fourth bar) and HQ with JUD (HQ-s,n,q, the last bar) are further compared in Table VI. The relative error rate reductions shown in the last row are significant and consistent for all SNR values, including the clean and 20-dB cases.

The above experimental results in Fig. 7 and Table VI are for a 4.4-kb/s bit rate. Further analysis was then performed for several better approaches found above with respect to different bit rates (4.4, 3.9, 3.3, and 2.7 kb/s) at all different SNR values. The results are shown in Fig. 8(a)–(f) for different SNR from clean to 0 dB, each with different bit rates. The four bars in each set in Fig. 8 are, respectively, for ECIVQ considering quantization error uncertainty for SVQ, the cascade of transform coding (TC) and ECIVQ (TC-ECIVQ), the cascade of HEQ and ECIVQ (HEQ-ECIVQ), and HQ with complete JUD including histogram shift (HQ-s,n,q). Here, except for the clean speech case at higher bit rates, HQ-s,n,q consistently performed

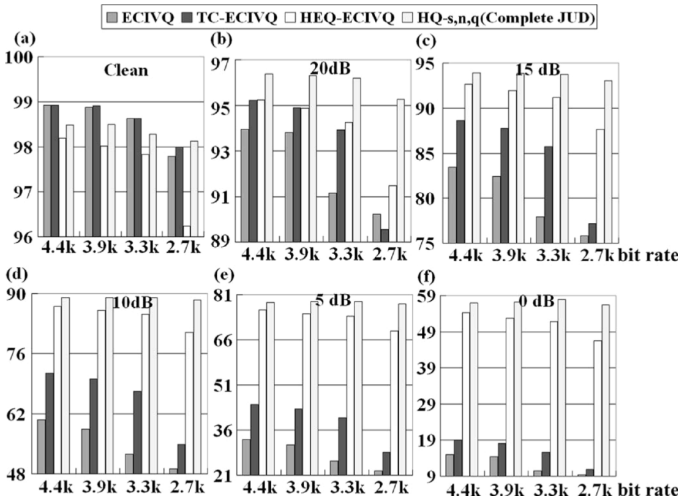

Fig. 8. Comparison of different approaches discussed in this paper for DSR (but without transmission errors) under different bit rates and SNR values. (a) Clean. (b) 20 dB. (c) 15 dB. (d) 10 dB. (e) 5 dB. (f) 0 dB.

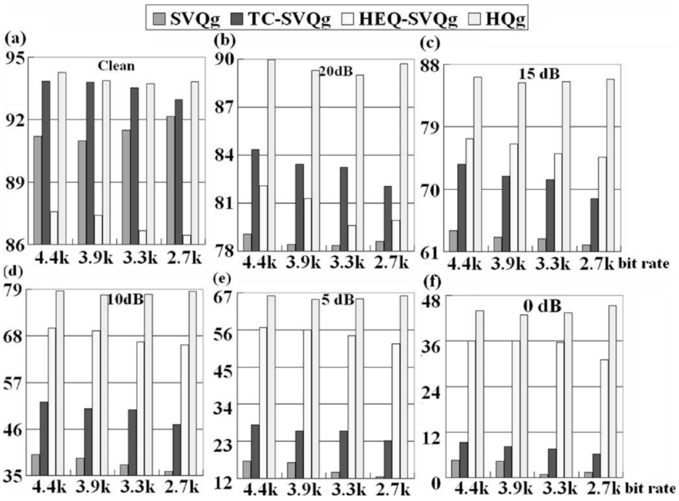

Fig. 10. Comparison of SVQg, TC-SVQg, HEQ-SVQg, and HQg (all with GPRS transmission errors), for different bit rates and SNR values. (a) Clean. (b) 20 dB. (c) 15 dB. (d) 10 dB. (e) 5 dB. (f) 0 dB.

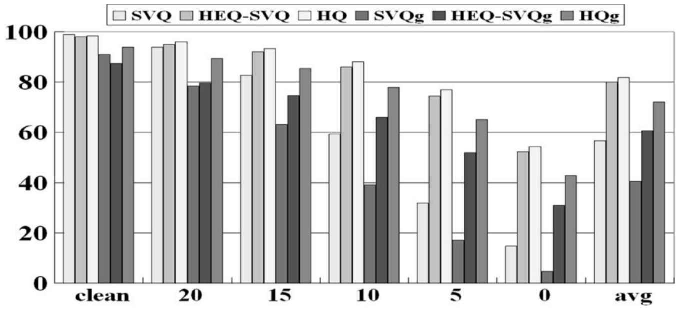

Fig. 9. Comparison of SVQ, HEQ-SVQ, and HQ, and those with GPRS transmission errors (SVQg, HEQ-SVQg, HQg), averaged over all types of noise, but separated for each SNR value.

better for all SNR values and all bit rates than other combinations of the front-end feature transformation (TC or HEQ) or back-end compensation considering quantization uncertainty (ECIVQ). Also, the performance of ECIVQ, TC-ECIVQ, and HEQ-ECIVQ are all more sensitive to lower bit rates, while HQ-s,n,q is relatively insensitive to different bit rates at all SNR conditions.

2) HQ-Based DSR Over Wireless Channels With Transmission Errors, But Without EC: We first compared the robustness of SVQ and HQ against environmental noise at the client end plus the transmission errors at a client traveling speed of 3 km/h, assuming no EC approach was used. Fig. 9 is the averaged results over all different types of noise but separated for different SNR values. The first three bars are the results for the standard SVQ, SVQ followed by HEQ front-end (HEQ-SVQ), and HQ (two-dimensional), all at 4.4 kb/s and without transmission errors, exactly the same as the first, third, and fifth bars in Fig. 7(b), and the next three bars are those suffering from GPRS transmission errors (SVQg, HEQ-SVQg, HQg: the label "g" indicates GPRS). For SVQ, the performance degradation caused by GPRS (first bar compared to fourth bar) is larger when SNR is lower, even with HEQ (second bar compared to fifth bar, e.g., 98.07% to 87.78% for clean speech, 91.97% to 76.74% for 15-dB SNR, and 85.86% to 68.73% for 10-dB SNR). Clearly, features corrupted by noise are more susceptible to transmission errors. The improvements that HQ offered over HEQ-SVQ when transmission errors were present (sixth bar to fifth bar) are consistent and significant at all SNR values. For example, in the case of 10-dB SNR with GPRS, HQ (sixth bar) offered an accuracy of 78.69% while the number was 69.84% for HEQ-SVQ (fifth bar). This verified that HQ is robust against both environmental noise and transmission errors.

The above results in Fig. 9 are for a 4.4 kb/s bit rate. Further analysis was then performed for several better approaches found above with respect to different bit rates (4.4, 3.9, 3.3, and 2.7 kb/s) for all SNR values (from clean to 0 dB) as shown in Fig. 10(a)–(f). The four bars in each set in Fig. 10 are, respectively, for SVQg, transform coding followed by SVQ (TC-SVQg), the cascade of HEQ and SVQ (HEQ-SVQg), and HQg, all with GPRS transmission errors. Here, HQ consistently performed better than different versions of SVQ enhanced by some feature transformation approaches (TC or HEQ) for all SNR values and all bit rates. With SVQ, features with environmental noise and quantization distortion are more sensitive to lower bit rates when transmission errors are present. For example, in the case of 5-dB SNR, the performance of HEQ-SVQ degraded from 56.66% at 4.4 kb/s to 51.88% at 2.7 kb/s. On the other hand, the performance of HQ is very stable for different bit rates in all cases of SNR, even with the presence of transmission errors. This verified that HQ is robust against not only quantization distortion and environmental noise, but transmission errors as well.

3) HQ-Based DSR Over Wireless Channels With EC: The next set of experiments tried to examine the effectiveness of the three-stage EC techniques for HQ proposed here in Section IV. Fig. 11 shows the results with GPRS transmission errors at a speed of 3 km/h, without and with the different EC approaches. The five bars in each set are, respectively, for SVQg, HEQ-SVQg, HEQ-SVQ with GPRS and with repetition (HEQ-SVQgr: the label "r" indicates the ETSI-recommended error mitigation strategy by repetition), HQg, and HQ with GPRS and the three-stage EC techniques propose here (HQgc:

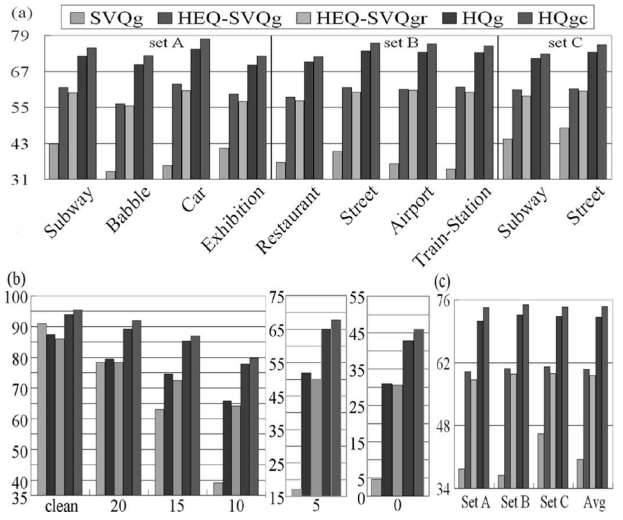

Fig. 11. Comparison of SVQ under GPRS (SVQg), HEQ-SVQ under GPRS without and with repetition (HEQ-SVQg and HEQ-SVQgr), HQ under GPRS without and with EC techniques (HQg and HQgc). (a) Averaged over all SNR values, but separated for different noise types in sets A, B, and C. (b) Averaged over all types of noise, but separated for each SNR value. (c) Averaged over all SNR values and noise types but separated for sets A, B, C.

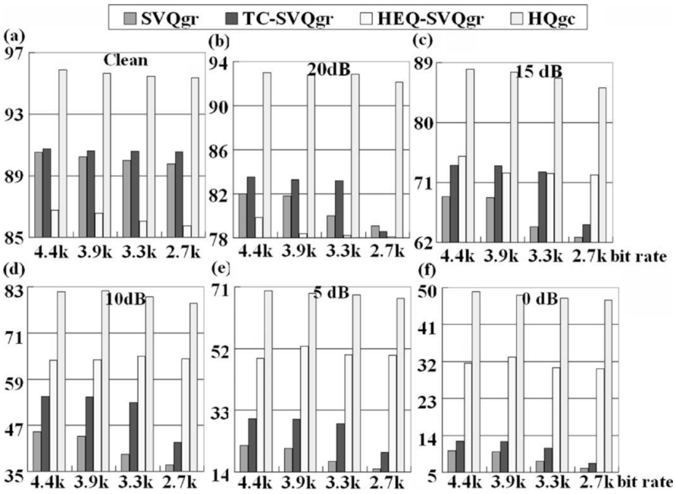

Fig. 12. Comparison of SVQgr, TC-SVQgr, HEQ-SVQgr (all under GPRS with repetition), and HQgc (under GPRS with error concealment) for different bit rates and SNR values. (a) Clean. (b) 20 dB. (c) 15 dB. (d) 10 dB. (e) 5 dB. (f) 0 dB.

the label "c" indicates three stage EC), all at bit rate of 4.4 kb/s. Fig. 11(a) are those averaged over all SNR values but separated for different noise types in sets A, B, and C, (b) are those averaged over all types of noise but separated for different SNR values, and (c) are those averaged over all types of noise and all SNR values but separated for sets A, B, and C. It can be found that the ETSI repetition technique actually degraded the performance of HEQ-SVQg (third bar versus second bar), probably because the whole feature vectors including the correct subvectors are replaced by estimations that are very possibly inaccurate. Under GPRS, HQg without any EC techniques (fourth bar) actually outperformed the first three bars for all cases. Applying the proposed three-stage EC techniques (HQgc, fifth bar) then further improved the performance significantly for all cases. This verified that the three-stage EC framework is robust against not only transmission errors, but against environmental noise as well.

The above results in Fig. 11 are for a 4.4 kb/s bit rate. Further analysis was then performed with respect to different bit rates (4.4, 3.9, 3.3, and 2.7 kb/s) for all SNR values as shown in Fig. 12(a)–(f). The four bars in each set in Fig. 12 are, respectively, for SVQ with GPRS errors and with repetition (SVQgr: the label "r" indicates the ETSI-recommended error mitigation strategy by repetition), TC-SVQ with GPRS errors and with repetition (TC-SVQgr), HEQ-SVQ with GPRS errors and with repetition (HEQ-SVQgr), and HQ with GPRS and the three-stage EC techniques propose here (HQgc). Here HQgc consistently performed better than all other approaches for all SNR values and all bit rates. For example, in the case of 10-dB SNR and a 3.3 kb/s bit rate, HQgc offered an accuracy of 81.57% compared to 38.92% for SVQgr, 53.34% for TC-SVQgr and 64.97% for HEQ-SVQgr. HQgc offered an accuracy of higher than 65% (67.42%) even at 5-dB SNR and the low bit rate of 2.7 kb/s. These indicate that HQ with the three-stage EC is robust against both environmental noise and transmission errors, and is insensitive to different bit rates.

The above results in Figs. 11 and 12 are for a client traveling at a speed of 3 km/h. We then consider other different client traveling speeds at 4.4 kb/s in Fig. 13. Here, the four cases shown in each figure are for HEQ-SVQ under GPRS, without and with ETSI repetition (HEQ-SVQg and HEQ-SVQgr), and HQ under GPRS, without and with the three-stage EC (HQg and HQgc), at traveling speeds of 3, 50, 100, and 250 km/h. Only two typical types of input speech noise, car for stationary and babble for nonstationary were taken as examples, since for some noise types such as exhibition or restaurant a client traveling speed above 3 km/h does not make sense. The results for two typical values of SNR, 15 dB and 5 dB plus those results averaged over all SNR values for car/babble noise are shown in Fig. 13(a1)/(a2)–(c1)/(c2), respectively. The superiority of HQ with EC (HQgc) is obvious as verified by the highest curves in all cases. As an example, for 15-dB car noise at 100 km/h as shown in Fig. 13(a1), the performance of HEQ-SVQ degraded seriously (78.74%), applying ETSI repetition on HEQ-SVQ did not help (72.89%), and HQ is much better (86.04%) while the three-stage EC offered very good improvements (92.80%). As another example, for 5-dB car noise as shown in Fig. 13(b1), the performance of HEQ-SVQ degraded seriously at high traveling speeds (e.g., 59.20% at 100 km/h); here, HQ was much better (e.g., 66.24% at 100 km/h), and the three-stage EC further improved the performance significantly (e.g., 78.29% at 100 km/h). On the other hand, as one more example in Fig. 13(a1) the HEQ-SVQ features with noise disturbances were more susceptible to higher transmission errors due to higher client traveling speeds (81.82% at 3 km/h and 78.74% at 100 km/h), while HQ features were more robust in this case (87.33% at 3 km/h and 86.04% at 100 km/h). This is why the curves for HQg are quite flat in almost all the six figures in Fig. 13, while those for HEQ-SVQg and HEQ-SVQgr decline faster as the client traveling speed increases. The curves for HQgc are also quite flat for car noise (Fig. 13(a1)–(c1)), but

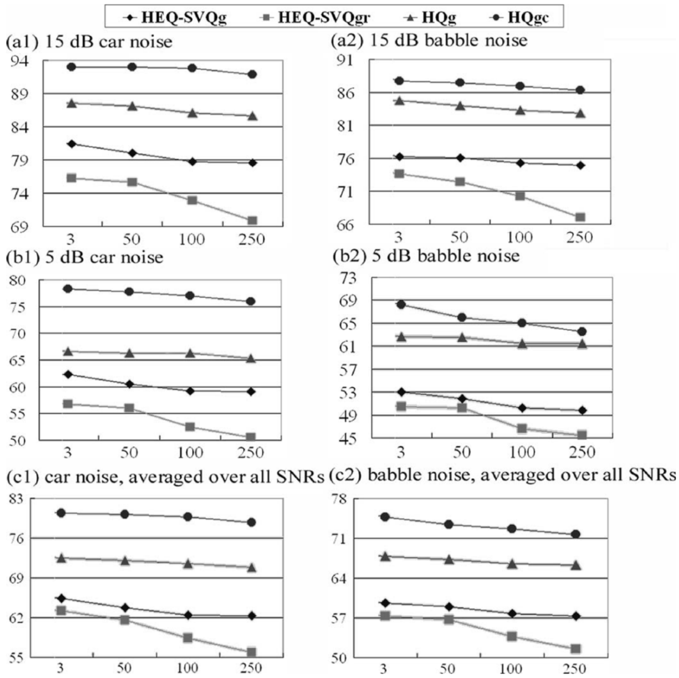

Fig. 13. Comparison of HEQ-SVQ under GPRS without and with repetition, HQ under GPRS without and with EC, at traveling speeds of 3, 50, 100, and 250 km/h. (a1)/(a2) for car/babble noise at 15-dB SNR. (b1)/(b2) for car/babble noise at 5-dB SNR. (c1)/(c2) for car/babble noise averaged over all SNR values.

less flat for babble noise [Fig. 13(a2)–(c2)]; the nonstationary nature of the babble noise is probably more difficult to handle with EC techniques.

VII. Conclusion

HQ is proposed in this paper, a novel approach for robust and/or DSR. HQ has been shown to be robust for all types of noise and all SNR conditions for either conventional speech recognitions systems, or DSR at all bit rates. The HQ configuration has been shown to be easily scalable based on bandwidth or noise conditions. For future personalized and context-aware DSR environments, HQ can be adapted to network and terminal capabilities, with recognition performance optimized based on environmental conditions. HQ can also provide more robust recognition features for many possible applications in the future.

Acknowledgment

The authors would like to thank the anonymous reviewers and associate editor for their extensive and valuable comments.

References

[1] V. Digalakis, L. Neumeyer, and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the world wide web," *IEEE Select. Areas Commun.*, vol. 17, no. 1, pp. 82–90, Jan. 1999.

[2] K. K. Paliwal and S. So, "Scalable distributed speech recognition using multi-frame gmm-based block quantization," in *Proc. ICSLP*, 2004, CD-ROM.

[3] J. A. Arrowood and M. Clements, "Extended cluster information vector quantization (ECI-VQ) for robust classification," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, May 2004, pp. 889–892.

[4] *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Extended Advanced Front-End Feature Extraction Algorithm; Compression Algorithms; Back-End Speech Reconstruction Algorithm*, , Nov. 2003, ETSI Std. ES 202 212 V1.1.1 Rec..

[5] B. Milner and X. Shao, "Low bit-rate feature vector compression using transform coding and non-uniform bit allocation," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Apr. 2003, pp. 129–132.

[6] Q. Zhu and A. Alwan, "An efficient and scalable 2D-DCT based feature coding scheme for remote speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2001, pp. 113–116.

[7] W.-H. Hsu and L.-S. Lee, "Efficient and robust distributed speech recognition (DSR) over wireless fading channels: 2D-DCT compression, iterative bit allocation, short BCH code and interleaving," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2004, pp. 69–72.

[8] S. Molau, M. Pitz, and H. Ney, "Histogram based normalization in the acoustic feature space," in *Proc. ASRU*, 2001, pp. 21–24.

[9] A. de la Torre, A. M. Peinado, J. C. Segura, J. L. Perez-Cordoba, M. C. Benitez, and A. J. Rubio, "Histogram equalization of speech representation for robust speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 3, pp. 355–366, May 2005.

[10] S. Chen and R. Gopinath, "Gaussianization," *Proc. Neural Inf. Process. Syst.*, pp. 423–429, 2000.

[11] I. Kiss and P. Kapanen, "Robust feature vector compression algorithm for distributed speech recognition," in *Proc. Eurospeech*, 1999, pp. 2183–2186.

[12] J. Droppo, A. Acero, and L. Deng, "Uncertainty decoding with SPLICE for noise robust speech recognition," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, 2002, pp. 57–60.

[13] J. A. Arrowood and M. A. Clements, "Using observation uncertainty in HMM decoding," in *Proc. ICSLP*, 2002, pp. 1561–1564.

[14] H. Liao and M. J. F. Gales, "Joint uncertainty decoding for noise robust speech recognition," in *Proc. Eurospeech*, 2005, pp. 3129–3132.

[15] N. B. Yoma, C. Molina, J. Silva, and C. Busso, "Modeling, estimating, and compensating low-bit rate coding distortion in speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 14, no. 1, pp. 246–255, Jan. 2006.

[16] C. Boulis, M. Ostendorf, E. A. Riskin, and S. Otterson, "Graceful degradation of speech recognition performance over packet-erasure networks," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 8, pp. 580–590, Nov. 2002.

[17] A. M. Peinado, V. Sanchez, J. L. Perez-Cordoba, and A. J. Rubio, "Efficient MMSE-based channel error mitigation techniques application to distributed speech recognition over wireless channels," *IEEE Trans. Wireless Commun.*, vol. 4, no. 1, pp. 14–19, Jan. 2005.

[18] A. Bernard and A. Alwan, "Low-bitrate distributed speech recognition for packet-based and wireless communication," *IEEE Trans. Speech, Audio Process.*, vol. 10, no. 8, pp. 570–579, Nov. 2002.

[19] A. Cardenal-Lopez, L. Docio-Fernandez, and C. Garcia-Mateo, "Soft decoding strategies for distributed speech recognition over ip networks," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, May 2004, pp. 49–52.

[20] V. Ion and R. Haeb-Umbach, "A unified probabilistic approach to error concealment for distributed speech recognition," in *Proc. Interspeech*, Sep. 2005, pp. 2853–2856.

[21] Z.-H. Tan, P. Dalsgaard, and B. Lindberg, "A subvector based error concealment algorithm for speech recognition over mobile networks," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, May 2004, pp. 57–60.

[22] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," in *Proc. ISCA ITRW ASR2000*, Sep. 2000, pp. 181–188.

[23] C.-Y. Wan and L.-S. Lee, "Histogram-based quantization (HQ) for robust and scalable distributed speech recognition," in *Proc. Interspeech*, Sep. 2005, pp. 957–960.

[24] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[25] J. Max, "Quantizing for minimum distortion," *IEEE Trans. Inf. Theory*, vol. 6, no. 1, pp. 7–12, Mar. 1960.

[26] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Speech Audio Process.*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[27] F. Hilger and H. Ney, "Quantile-based histogram equalization for noise robust speech recognition," in *Proc. Eurospeech*, 2001, pp. 1135–1138.

[28] C.-Y. Wan and L.-S. Lee, “Joint uncertainty decoding (JUD) with histogram-based quantization (HQ) for robust and/or distributed speech recognition,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, May 2006, pp. 125–128.

[29] C.-Y. Wan, Y. Chen, and L.-S. Lee, “Three-stage error concealment for distributed speech recognition (DSR) with histogram-based quantization (HQ) under noisy environment,” in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Process.*, Apr. 2007, pp. 877–880.

[30] B. Milner and A. James, “Robust speech recognition over mobile and IP networks in burst-like packet loss,” *IEEE Trans. Speech Audio Process.*, vol. 14, no. 1, pp. 223–231, Jan. 2006.

[31] *Subjective Performance Assessment of Telephone-Band and Wideband Digital Codecs, Annex D: Modified IRS Send and Receive Characteristics*, , Feb. 1996, ITU-T Std. ITU-T Rec. P.830.

[32] J.-H. Chen, “Receiver design and simulation analysis of GPRS physical layer,” M.S. thesis, National Taiwan Univ., Taipei, 2001.

[33] C.-P. Chen and J. A. Bilmes, “MVA processing of speech features,” *IEEE Trans. Speech Audio Process.*, vol. 15, no. 1, pp. 257–270, Jan. 2007.

[34] J.-W. Hung and L.-S. Lee, “Optimization of temporal filters for constructing robust features in speech recognition,” *IEEE Trans. Speech Audio Process.*, vol. 14, no. 3, pp. 808–832, May 2006.

[35] H. Hermansky, “Trap-tandem: Data-driven extraction of temporal features from speech,” in *Proc. ASRU*, 2003, pp. 255–260.

Chia-Yu Wan (S’06) received the B.S. and M.S. degrees in electrical engineering from National Taiwan University (NTU), Taipei, Taiwan, R.O.C., in 2003 and 2005, respectively. She is currently pursuing the Ph.D. degree in the Speech Processing Laboratory, Graduate Institute of Communication Engineering, NTU.

Her research interests are primary on distributed speech recognition and robust speech recognition.

Lin-Shan Lee (F’93) received the Ph.D. degree in electrical engineering from Stanford University, Stanford, CA.

He has been a Professor of electrical engineering and computer science at the National Taiwan University, Taipei, Taiwan, R.O.C., since 1982 and holds a joint appointment as a Research Fellow of Academia Sinica, Taipei. His research interests include digital communications and spoken language processing. He developed several of the earliest versions of Chinese spoken language processing systems in the world including text-to-speech systems, natural language analyzers, dictation systems, and voice information retrieval systems.

Dr. Lee was Guest Editor of a Special Issue on Intelligent Signal Processing in Communications of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS in December 1994 and January 1995. He was Vice President for International Affairs (1996–1997) and the Awards Committee chair (1998–1999) of the IEEE Communications Society. He was a member of Permanent Council of International Conference on Spoken Language Processing (ICSLP), is currently a member of the Board of International Speech Communication Association (ISCA 2002–2009), a Distinguished Lecture of the IEEE Signal Processing Society (2007–2008), and will be the general chair of ICASSP 2009 in Taipei.