# Graceful Degradation of Speech Recognition Performance Over Packet-Erasure Networks

Constantinos Boulis, Mari Ostendorf, *Senior Member, IEEE*, Eve A. Riskin, *Senior Member, IEEE*, and Scott Otterson

***Abstract*—This paper explores packet loss recovery for automatic speech recognition (ASR) in spoken dialog systems, assuming an architecture in which a lightweight client communicates with a remote ASR server. Speech is transmitted with source and channel codes optimized for the ASR application, i.e., to minimize word error rate. Unequal amounts of forward error correction, depending on the data's effect on ASR performance, are assigned to protect against packet loss. Experiments with simulated packet loss in a range of loss conditions are conducted on the DARPA Communicator (air travel information) task. Results show that the approach provides robust ASR performance which degrades gracefully as packet loss rates increase. Transmitting at 5.2 Kbps with up to 200 ms added delay, leads to only a 7% relative degradation in word error rate even under extremely adverse network conditions.**

***Index Terms*—Bit allocation, forward error correction, packet loss, speech recognition, unequal loss protection.**

## I. Introduction

AUGMENTING the features of low power, portable, hand-held devices with automatic speech recognition (ASR) capability is an area of increasing interest. ASR capability is considered to be more important to hand-held devices than to desktop computers because of the absence of easy-to-use human-computer interaction devices such as a keyboard and mouse. With ASR, users of hand-held devices will be able to make travel arrangements, receive weather and traffic updates, and retrieve information from large databases without the assistance of a human operator.

One approach toward this goal is to integrate a speech recognizer on the device and perform recognition locally. For simple tasks, such as voice dialing or controlling embedded systems, this is a good strategy. However, for more complex tasks such as spoken dialog systems, it makes more sense to transmit coded speech and perform ASR on a remote server [1]–[4]. One reason is that most hand-held devices are low power, and recognizers used in dialog systems need high computational resources. However, even as hardware becomes more powerful, the placement of the ASR computation on a remote server offers the advantage that the ASR system can more easily be adapted according to the information state of the back-end, which allows for dynamic vocabularies and language models. Another advantage of this approach is that updates of speech recognizers need to be done only on the server side, rather than having every device updated. This results in a much faster and cost-effective update procedure, and one that is effectively "invisible" to the client/customer.

In the client-server architecture envisioned here, the low power hand-held device will perform limited computation operations (such as feature extraction and quantization), and then will transmit the data to a distant server which has the computational power to carry out speech recognition and understanding. Before deploying such systems, there are two major issues that need to be addressed. The first issue is that the data are transmitted over an unreliable communication channel, which results in loss of data and hence degradation of performance. The problem can be addressed by introducing forward error correction and/or error concealment techniques, as in speech coding, but it should be specifically tailored for ASR. The second issue is that the time between when a user speaks a sentence and receives the result (termed total delay) is now increased by the round-trip transmission time between the client and the server. If we introduce error correction techniques, we have to interleave data which adds another delay (termed data acquisition delay). It is known that if the total delay is more than 500 ms, human-computer interaction becomes uncomfortable [5].

A first approach to the client-server model is to use a speech codec (encoder-decoder) to transmit voice over a communication channel and then use a standard procedure to extract features and perform recognition. However, several authors have shown that this approach results in significant performance loss relative to using features computed from uncoded speech, even when the channel is noiseless [6]–[9].

A second approach is to train an ASR system based on the speech codec signal itself rather than on the original waveform. For different versions of the GSM speech codec, this approach offers improvement over the first approach for word recognition [10]–[12], but not for speaker recognition [13]. Alternative source coding methods designed for both audio and ASR applications are proposed in [14], [15].

A third approach is to perform the feature extraction method locally, quantize the features, and then transmit the codewords of the features over the channel. Other work has shown that substantial bit rate reductions are possible with little or no degradation in ASR performance relative to the unquantized case, with reported bit rates of 4.0 Kbps for several languages [16] and 2.0 Kbps for a small vocabulary task [3]. This is clearly an advantage over approaches that build on GSM coding standards,

Manuscript received October 15, 2001; revised August 14, 2002. This work was supported by DARPA under Contract N660019928924 and NSF Grants EIA-9973531 and CCR-0104800. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Harry Printz.

The authors are with the Department of Electrical Engineering, University of Washington Seattle, WA 98195 USA (e-mail: boulis@ee.washington.edu; mo@ee.washington.edu; riskin@ee.washington.edu; scotto@ee.washington.edu).

Digital Object Identifier 10.1109/TSA.2002.804532

where the total bit rates range over 5.3–13 Kbps. A disadvantage for wireless phones is that an extra signal coding component (in addition to the GSM speech codec) has to be added to the client, which is not the case for approaches that work directly with the coded speech. However, the cost of a second encoder is small, relative to a full recognizer.

Interest in this third approach is growing, with complementary threads of research on source and channel coding. In source coding, the focus has been on the choice of features for coding. In [17], the Mel-frequency cepstrum coefficient (MFCC) vectors are grouped into blocks, a 2-D discrete cosine transform (DCT) is applied to a group of vectors, and the DCT coefficients with very small energy are ignored. Good performance is achieved on the TI digit task for rates as low as 624 bps, even in moderate noise conditions. When quantizing features derived from line spectral pairs, rates as low as 300 bps are found to be sufficient for good digit recognition in [18]. If fewer bits are needed for encoding feature vectors, then it is possible to add more bits for error coding, as also shown in [18], which is made more effective by using a soft (rather than hard) decoding scheme for reconstructing the feature vectors. Soft decoding schemes that operate within the recognition system are proposed in [19], [20], where the probability of observing a frame is weighted by the confidence of correctly receiving each bit. Both systems use forward error correction to minimize the impact of channel bit errors. In [19], an unequal loss protection scheme is introduced, but the total bit rate is high because of the use of scalar quantization. (The first MFCCs are protected more than the last ones, since they are more important to ASR performance in that they describe the overall spectral shape.) In [20], which uses a linear predictive VQ coding of MFCCs, good performance on the Aurora task is achieved with a 1 Kbps rate (including error correction bits) evaluated on a Rayleigh fading channel. All these proposals use reconstructed feature vectors in a recognition system based on continuous distributions, as we shall also do in this paper. Another line of research that complements all of this work is the use of the coded speech directly in a system based on discrete distributions [21].

The work in [18]–[20] increases robustness to error-prone communication channels; however, the emphasis is on transmitting a continuous bitstream. Many modern communication channels are packet-switched, meaning that bits are organized into packets and then transmitted. For example, the Internet is a widely deployed network of computers that allows the exchange of data packets. When the number of packets sent exceeds transmission capacity, packets are discarded at random, causing loss of data and potentially decoding failure if the lost data are not retransmitted. Each packet can be assigned a unique sequence number, so it is known which packets are received and which are lost.

Packet loss has been studied in the context of speech and audio coding [22]–[26] but to date, there has been little work on the effects of packet loss on ASR. Notable exceptions are [12], [27], [28], which use a simple interpolation scheme for error concealment. The experimental results are very good, but they are based on an assignment of one cepstral vector per packet which is very inefficient. Further, much of the above work is carried out on limited size vocabularies (isolated words or connected digits), and it is unclear how the results scale when applied to a larger vocabulary task. In this work, we address packet loss in ASR using more practical packetization schemes and evaluate performance on a medium-size vocabulary task used in a human-computer dialog system. An important contribution is an unequal loss protection (ULP) algorithm to assign forward error correction (FEC) to minimize word error rate (WER) in an ASR task.

This paper is organized as follows. The overall system architecture and recognition task are described in Section II, and the source coding techniques we use are explained in Section III. In Section IV, we describe our forward error correction and concealment methods, and the ULP algorithm for assigning FEC to data is outlined in Section V. In Section VI, we present experimental results for various simulated loss conditions, and in Section VII we summarize the findings and point out possible directions for future work.

## II. ASR System

The goal of this work is to develop a source and channel coding scheme designed explicitly for minimizing word error rate, with a design criterion that no changes will be required in the speech recognition system at the server (after feature vector reconstruction).[1] Since the recognizer is designed for a particular front end, we have not explored alternative features and use the standard feature set for this recognizer (energy, 8 MFCCs and associated derivatives). Only the cepstrum coefficients are quantized and transmitted; the delta and delta-delta features (derivatives) are computed on-line at the receiver and appended at the reconstructed frame.

A block diagram for the proposed system is shown in Fig. 1. We start with a standard acoustic processor, then quantize the feature vectors using an embedded code with fixed-length codewords, designed to minimize word error rate. Next, a ULP algorithm assigns FEC to minimize expected WER and then all data and FEC bits are packetized. At the receiver, the packets are decoded, resulting in a bit sequence that is sent to the VQ decoder. If more packets are lost than can be corrected with the channel code used, the VQ decoder simply uses a lower rate code. The output of the VQ decoder is a reconstructed feature vector that can be passed to a standard ASR decoder. Optionally, we can use an error concealment step in addition to forward error correction. Each of the components is further described in the sections to follow.

As it will be useful to present intermediate experimental results, we briefly describe the experiment paradigm here. All the experiments were carried out on a corpus of speech from human-computer dialogs about air travel information, a task associated with the DARPA Communicator program. The data were collected over the telephone. To evaluate our algorithms, we used a continuous density HMM-based recognizer and telephony acoustic models provided to us by Nuance. The vocabulary size was 2647 words. We used a standard trigram language model, trained with about 50 000 sentences of Communicator

[1]While one can clearly better optimize ASR performance by also changing the recognizer, we would lose some of the advantage of easy upgrades/updates to the recognition system if it is tuned to the coding system.

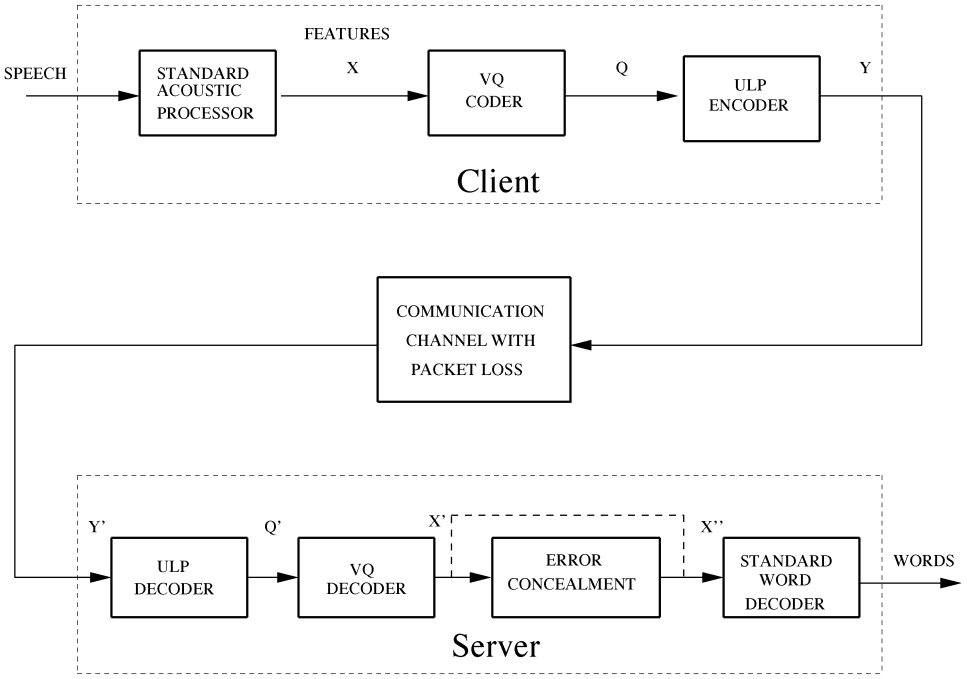


Fig. 1. Block diagram for proposed system.

data, collected at the University of Colorado [29] and Carnegie Mellon University (CMU) [30]. The codebooks for the vector quantization are estimated using Euclidean distance as the optimization metric and 20 K sentences as training data (training set). For the source coder, the number of bits assigned to each subvector is carried out using WER as the optimization criterion on 875 sentences, 2096 words (development set, collected at CMU). The same development set is used for the ULP design using expected WER as the optimization criterion, to assign FEC bits to data. The baseline WER for the development set is 22.9%, comparable to that reported in [29]. A third data set of 2100 sentences, 6200 words, collected at the University of Washington, is used for an independent evaluation. The baseline performance for the evaluation set is 24.5% WER.

## III. Source Coding

### A. VQ Alternatives

Source coding is performed by applying vector quantization (VQ) [31] to the MFCC vectors. We use product code VQ [32], where the subvectors are individually quantized using binary tree-structured vector quantization (TSVQ) [33] with fixed length coding. The binary structure will prove to be useful for efficiently finding the tradeoffs between bit rate and WER, as described further in Section III.B. We use Euclidean distance as the distortion criterion for training the codebooks [31], although WER is used for estimating the codebook size for each subvector, as described in more detail in Section III.B. We have experimented with two variations—intra-frame VQ and inter-frame VQ—described below.

In intra-frame VQ, we group the coefficients of each MFCC vector into sets and independently quantize each subvector. Intuitively, for VQ to be most effective, each subvector must have coefficients which are maximally dependent on each other. In addition, because we separately VQ each subvector, the subvectors should be as independent as possible. In general, the problem of determining the set of subvectors is related to unsupervised clustering. In [3], correlation-based and knowledge-based partitioning techniques have been investigated, with the best results obtained from grouping consecutive coefficients (knowledge-based approach). We also investigated use of pairwise mutual information, but did not get any gain over the simple adjacency-based grouping. Our best results were obtained by using 5 subvectors with the cepstrum coefficients, as shown in Table I with the associated bit allocations, which is similar to the configuration used in [3]. On the development set, we find that the cepstrum parameters can be quantized at a rate of 2.6 Kbps and have no degradation in WER relative to using the original (uncoded) vectors, though a small degradation is observed on the evaluation set (from 24.5% to 24.8% WER).[2] This is roughly comparable to the result in [3], which found 2.0 Kbps was needed for a similar task with a smaller vocabulary (1500 words).

TABLE I
Cepstrum Coefficients and Associated Number of Bits Assigned to Each Subvector in Intra-Frame Coding at 2.6 Kbps

| Subvector | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| MFCCs | {1,2} | {3,4} | {5,6} | {7,8} | {E} |
| # bits | 6 | 6 | 4 | 6 | 4 |

In inter-frame VQ, we exploit the very high dependencies that exist between the same cepstrum coefficients of neighboring frames. In pilot experiments, we observed that the mutual information between the same coefficients of neighboring frames is

[2]Increasing the training set size from 20 K to 50 K sentences, gave almost identical results.

TABLE II
CEPSTRUM COEFFICIENT INDEX AND NUMBER OF BITS ASSIGNED TO EACH 2-DIMENSIONAL SUBVECTOR (MFCCS FROM CONSECUTIVE FRAMES) IN INTER-FRAME CODING AT 1.2 Kbps

| Subvector | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| MFCCs | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | E |
| # bits | 4 | 3 | 3 | 3 | 2 | 3 | 3 | 4 | 5 |

5–10 times higher than the mutual information between neighboring coefficients of the same frame. Thus, we form 9 clusters where cluster $i$ has cepstrum coefficient $i$ from each of 2 consecutive frames. The subvectors have dimension 2, since that provides a close match to the intra-frame VQ configuration. As expected, inter-frame VQ is much more efficient, allowing a rate of 1.2 Kbps with no degradation in WER relative to the original vectors. The bit allocation for this scheme is given in Table II. Comparing the allocation for the blocks that are common across the two schemes, the bit allocations are proportionally the same, as expected.

### *B. Bit Allocation Driven by WER*

For each of the source coding methods described, we need to determine the impact of losing each bit on the WER. Commonly in TSVQ, the distortion measure used is the mean-squared error. Since this is an additive distance, given an initial TSVQ for the cepstrum coefficients, one can efficiently find an optimal curve indicating the tradeoff of distortion versus bit rate [34]. However, in our case the objective is minimum WER and not minimum distortion. Since WER is not additive, the efficient algorithm does not apply. Instead, obtaining the optimal WER/bit rate tradeoff would require evaluating all possible combinations of deleting bits and picking the one with the lowest WER. If there are $M$ subvectors and each has $b_i$ bits in the full allocation ($\sum_{i=1}^{M} b_i = B$), then there are $B!/(\prod_i b_i!)$ possible combinations. Starting with $B = 40, M = 5$ and $b_i = 8$, there would be more than $10^{24}$ combinations. Thus, an exhaustive search would demand huge amounts of processing time, especially since a full recognition experiment is required each time any bits are deleted. Therefore, we experimented with various search space reduction techniques which are orders of magnitude less expensive, assessing the correlation of each with WER.

Although WER is not an additive distance, we still use a pruning strategy, i.e., at each step remove the bit that results in the smallest possible increase in WER. In the pruning strategy, we start with $B$ bits total, chosen according to some reasonable heuristic, and then eliminate one bit at each step. This is in contrast to a growing strategy that finds the best bit to add to decrease WER (as in the bit allocation assignment used in [3]). We conjecture that the search space reduction techniques will be less costly by using pruning rather than a greedy growing strategy, since for additive distances optimal pruning gives improved performance over greedy growing alone [34]. In addition, a pruning strategy will result in smaller approximation errors at high rates than at low rates, and it is the high rates at which we hope to most frequently operate.

There were two potentially complementary methods used to reduce the search cost. The first was to eliminate candidates at each step using a lower cost criterion, and evaluate using the higher cost WER on this subset. The second was to evaluate candidates at each step using a lookahead strategy that finds the best way to remove $K$ bits before deciding which last bit to remove. Thus, given $M$ subvectors, each of the $B$ steps will require roughly $M^K$ evaluations, assuming $K$ is relatively small.

Initially, we tried using $K = 1$ with three different low cost criteria: Euclidean distance, Mahalanobis distance using the global covariance matrix, and the acoustic likelihood of the training data. The two distances compared the reconstructed feature vector sequence to the original feature vector sequence. Being additive, these distances can be computed very quickly, changing only the contribution associated with the leaf of the tree affected by the candidate bit removal. The acoustic likelihood is computed from a forced alignment procedure, which is much more expensive than the distance measures but still a factor of 10 less costly than running recognition and much more effective than the distance measures. With $K = 1$ and $M = 5$, there are only 5 alternatives to evaluate at each step, so reducing these to a smaller set (3, in our experiments) before running WER evaluation did not provide sufficient computational savings given that there was a degradation in performance relative to evaluating the full set of 5 alternatives.

Next, we investigated use of WER with lookahead to determine whether the lookahead was in fact useful. We experimented with $K = 1, 2, 3, 4$, but the resulting curves were found to be almost identical. Since it did not appear that lookahead was useful even with the expensive WER criterion, we did not try combining the low cost search space reduction techniques with this approach. All remaining experiments used single bit removal with no lookahead (i.e., with $K = 1$) based on the WER criterion. We also experimented with different starting points (4 Kbps and 2.8 Kbps), and the results were insensitive to these changes, too. For the intra-frame VQ case with $M = 5$ this requires reasonable computational resources. For the inter-frame VQ case, there are $M = 9$ subvectors and because each time we delete a bit we decrease the mean bit rate by only 0.5 bits/frame, the number of steps is doubled. This increases the total computing time to obtain the new WER/bit-rate curve by a factor of 4. With further increases in the computation, it may be worth revisiting the use of likelihood for search space reduction.

The result of this procedure is a curve (or table) showing WER as a function of the number of bits and an array identifying the specific bit that is deleted at each step. This information is stored and used as a basis to assign different numbers of source and channel coding bits to each subvector, as described in the next section. Clearly, if losing bit $i$ increases the WER more than losing bit $j$ then bit $i$ should be assigned more FEC.

The WER curves for the two different quantization methods are shown in Fig. 2. We observe that the WER performance of intra-frame VQ degrades rapidly for bit rates lower than 15 bits/frame. On the other hand, the performance of inter-frame VQ is still good even at much lower bit rates.

## IV. HANDLING PACKET ERASURES

In packet network transmission scenarios, the main problem is erasures of whole packets; bit errors can also occur but their

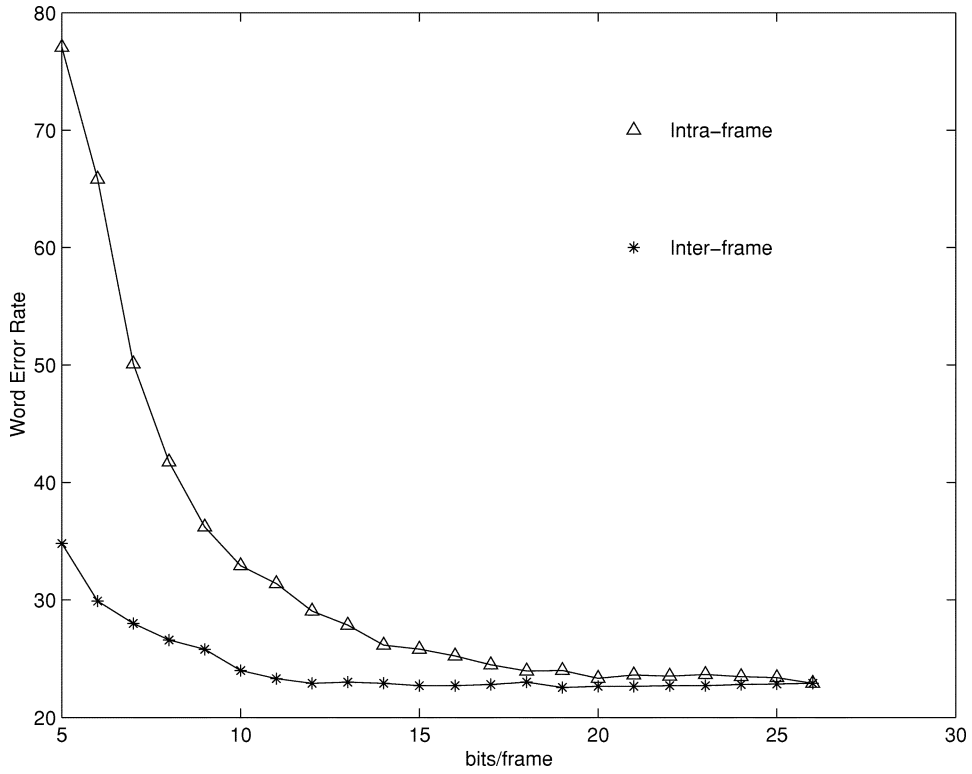


Fig. 2. WER/bit rate trajectories for intra- and inter-frame VQ.

probability is much lower. One can address the problem of data loss either by adding redundancy to the signal to be transmitted using a channel code (as in forward error correction), or by exploiting redundancy in the transmitted signal at the receiver in reconstructing the signal (as in error concealment). Both methods and their combination are explored here.

**Forward error correction**, such as Reed-Solomon coding [35], protects against erasures in data sent over lossy packet networks such as wireless networks or the Internet. In an $(N, k)$ Reed-Solomon code, $N-k$ symbols of FEC are used to protect $k$ symbols of data, meaning that $N$ symbols of data in total are transmitted. As long as any $k$ of the $N$ symbols are received, the $k$ original symbols of data can be recovered. Note that any error correction code can be used here, with a tradeoff associated with the power of the code versus added processing delay and overhead. Reed-Solomon codes are used in most of the experiments here, because they are convenient to analyze and are flexible. However, we also include experiments with a simple repetition code that gives results with a very low latency at the expense of a higher and less flexible rate.

In [36], a ULP algorithm was designed to assign unequal amounts of FEC to compressed image data. The ULP algorithm was designed for embedded codes, i.e., codes which allow intermediate reconstructions of the image from any prefix of the bitstream, and hence the ULP algorithm always assigns more FEC to the early parts of the coded bitstream. As a result, there is graceful degradation of compressed image quality with increasing packet loss; if a large amount of packet loss occurs, the decoded image is simply reconstructed from a shorter prefix of the coded bitstream, leading to reduced image quality instead of catastrophic failure.

TABLE III
ALLOCATION OF BITS IN EACH REED-SOLOMON CODE (R-S STREAM) TO DATA, FOR THE INTRA-FRAME VQ AT 4.8 Kbps, 200 ms DELAY AND A MEAN PACKET LOSS RATE OF 38.5%. NUMBERS INDICATE THE SUBVECTOR THAT DATA BITS CORRESPOND TO, AND "$F$" INDICATES AN ERROR CORRECTION BIT

| R-S stream | Packets (1-12) | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | 2 | 3 | 4 | F | F | F | F | F | F | F | F |
| **2** | 5 | 1 | 1 | 3 | F | F | F | F | F | F | F | F |
| **3** | 5 | 1 | 2 | 2 | 3 | 4 | F | F | F | F | F | F |
| **4** | 4 | 5 | 4 | 1 | 1 | 4 | 3 | 4 | 2 | 2 | 2 | 5 |

When speech MFCC vectors are encoded using tree-structured VQ [33], the code is also embedded; thus, the ULP algorithm can be conveniently applied to ASR data. The open question is how much FEC should be assigned to the MFCC vectors. Whereas the work in [36] optimized a peak signal-to-noise ratio, here we assign FEC to minimize the WER. The assignment of FEC avoids catastrophic failure by gradually reducing the quality of the reconstructed speech vectors. Similar to the image coding work, when packet loss occurs, the WER performance will degrade gracefully instead of falling off sharply. As a result, useful ASR is possible even in cases of high packet loss rates.

Table III shows how unequal loss protection is assigned for the case of intra-frame coding, where the data rate is 2.6 Kbps and the coded data is transmitted at 4.8 Kbps and 200 ms delay. The data bits are assigned to different codes (referred to here as R-S streams) according to their relative importance. In this case, bits 1–4 are assigned to the first R-S stream, bits 5–8 to the second, bits 9–14 are assigned to the third, and the rest are

assigned to the fourth R-S stream. The algorithm for allocating different numbers of error correction bits—or equivalently, adjusting the power of the Reed-Solomon code—is described in the next section.

An alternative to forward error correction is **error concealment**. One of the simplest methods of error concealment is interpolation [27], [28], [12]: when a frame is lost, it is linearly interpolated using the neighboring frames. The weight of each one of the neighbors is a linear function of the distance of the missing frame from its most recent correctly received neighbors. Here, we use a similar strategy (with $\pm 2$ nearest received frames), but unlike previous work, we assign more than one frame to a packet, so losing a packet results in the loss of more than one consecutive frame. As a consequence, the interpolation is based on more distant frames and is therefore a poorer representation of the original vector.

For the application of speech recognition (versus coding), there is yet a third alternative for dealing with erasures, which is to treat these as missing features. This involves simply dropping the observation term from the Viterbi update equation. In [14], it is shown that the missing feature approach gives performance that is similar to error concealment using a causal estimator.

As we see in Fig. 1 the error concealment module is applied on the server side and can be independent of the source and channel coding method implemented. Therefore, we can potentially gain further improvements by combining FEC and error concealment.

## V. ULP Assignment Algorithm

In this work, the approach to ULP assignment involved two steps. First, we identified the possible code configurations in terms of the size (in packets) of each data/FEC "ensemble," i.e., the number of packets $N$ that are grouped together for the $(N, k)$ Reed-Solomon block code, and the number of bits $b$ in each data/FEC symbol. The number of bits per symbol determines the number of R-S streams $S$, which is an upper bound on the number of available protection levels. Given a particular configuration, we find the value of $k_i$ for each R-S stream $i$ that optimizes speech recognizer performance.

### A. Packetization Constraints

In this work, we packetized our speech recognition data into packet sizes and data rates found in Internet voice over IP (VoIP). Specifically, we chose 10 ms frames and 10 byte packets, as found in G.729, and limited the total data rate $R$ (i.e., the speech data plus FEC symbols) to a total of 4.8 Kbps, as in some LPC vocoders. For a few comparative experiments, we also used rates of 2.6 Kbps and 5.2 Kbps.[3] Having fixed these parameters, the possible code configurations are limited by imposing the following constraints:

a) The ensemble contains $L$ MFCC vectors, where $3 \leq L \leq 20$ to reduce header overhead and limit added latency. (For $L = 20$, used in many experiments here, the added coding delay is roughly 200 ms.)
b) The number of packets in the ensemble is determined by the target data rate, the packet payload, the MFCC analysis rate, and the number of MFCC vectors $L$, since $R = 80N/(.01)L$ which gives $N = RL/8000$.
c) All FEC/data symbols are the same size ($b$ bits), where $b \geq \log_2 N$. The minimum size restriction is due to the use of Reed-Solomon codes. The same-size restriction is not strictly necessary but it limits the search space of possible packetizations.

In most experiments, we further restrict the possible choices, assuming that each data symbol consists of the same bit from all the frames of the ensemble, so $b$ is constrained to be the same as the number of frames of speech. Thus, an ensemble of 20 frames would have data symbols of 20 bits each. If a data symbol is lost then the same bit is lost from all frames of the ensemble. The motivation for this choice, besides simplicity, is that it is consistent with our criterion for ULP optimization (described next): if we lose a bit, we lose the same bit from all frames. While it may be useful to look at other symbol sizes (e.g., $b \in \{4, 5, 10\}$ for a 20-frame ensemble), our exploratory experiments varying $b$ did not show significant performance gains.

To illustrate how these constraints determine the ensemble characteristics, consider the case where the combined data and channel coding rate is 4.8 Kbps. If the maximum latency is used, then there are $N = 12$ packets, the symbol size is $b = L = 20$, and there are $S = 80/b = 4$ streams. Decreasing the rate would change the number of packets, but the other parameters would not change. Keeping the 4.8 Kbps rate but reducing the latency to 50 ms results in $N = 3, b = 5$, and $S = 16$. The smaller $N$ and $b$ mean that the error protection code is less powerful than for the maximum latency case.

### B. FEC Symbol Assignment

Having defined all possible ensembles, the next task is to determine the code with the minimum expected WER; i.e., for each ensemble configuration (length $N$, symbol size $b$), find the data allocation to R-S streams $(k_1, k_2, \ldots)$ and its associated cost (discussed next), and then choose the ensemble with the lowest cost. Without loss of generality, and consistent with [36], we can restrict the set of possible FEC assignments for a particular ensemble configuration to those for which the number of FEC symbols per row was nondecreasing. The most important bits (lowest number on the WER/bit rate tradeoff curve) are assigned to the first row, the next most important to the next row, etc. Thus, the set of possible FEC assignments are given by the set of possible cut points on the WER/bit rate tradeoff curve with nondecreasing lengths in bits.

Ideally, the optimization would be based on a bursty channel model, which could be evaluated by running ASR with a Monte Carlo channel loss model and iteratively testing candidate FEC assignments. Since this is computationally prohibitive, we instead estimate the expected WER for a given FEC allocation by using the WERs measured during bit allocation that assumes a steady loss rate. (We conduct the final evaluation using a more

[3]Note that 10 bytes is the payload; the header/trailer can require up to 62 bytes, depending on the generality required. If we only operate on UDP, we need only a 7 byte header (including the packet ID), in which case the 4.8 Kbps rate becomes 8.16 Kbps total. With one MFCC vector per packet, the data rate is 13.6 Kbps.

realistic channel model.) In this case, the FEC assignment is chosen by minimizing

$$\left(\sum_{k=1}^{N} p(k)\right)^{-1} \left(\sum_{k=1}^{N} p(k)\mathcal{E}(k)\right) \tag{1}$$

where $N$ is the number of packets in the ensemble (which can vary with the different ensemble candidates), $p(k)$ is the probability of losing $k$ out of $N$ packets according to the Poisson model, and $\mathcal{E}(k)$ is the word error rate when $k$ packets are lost. (Note that, with the Reed-Solomon code, only the total number of packets lost and not the particular combination of lost packets, determines the number of bits lost.) Given that $k$ packets are lost, we can identify the number of bits lost for a particular FEC assignment and therefore use the WER/bit tradeoff curve to obtain the relevant WER. Equation (1) corresponds to the expected WER under a truncated version of the Poisson distribution and assuming that a loss of each bit occurs uniformly for all vectors in the speech sample. Both assumptions are approximations, and the result is that the error rate is under-estimated by 3–4%.

Since the number of candidate configurations and ensemble size was small, it was possible to search over all FEC assignments for the assignment that yielded the lowest estimated expected WER. For example, for the case with 4.8 Kbps, intra-frame VQ and 200 ms delay, there are 68 candidate configurations.

## VI. Channel Simulation Experiments

Since the Poisson distribution used in the ULP design is not a very good model of network behavior, the resulting expected WER is not a reliable indicator of performance. Therefore, we conducted word recognition experiments simulating a more complex loss function. In the sections to follow, we describe the channel model used, the baselines for comparison, and results with obtained under different rates and latencies for error protection versus error concealment. All results are based on the 10 byte packet standard, and assume that the position of the missing packets can be identified by the sequence ID number which is included in the header of each packet.

### A. Channel Model

Most real communication channels exhibit burstiness. Such channels can be modeled by a 2-state Markov model [41], also known as a Gilbert model. In Fig. 3, $p$ is the probability that the next packet is lost, provided the previous one has arrived; $q$ is the probability that the next packet is not lost, given that the previous one was lost. If $p + q = 1$, the Gilbert model reduces to a Bernoulli model. The parameter $q$ can be seen as controlling the burstiness of packet losses. Although our FEC assignment was optimized with an exponential packet loss probability mass function, we ran tests under the loss conditions reported in [42], [41]. These conditions are summarized in Table IV, where $\mathrm{mlp} = p/(p+q)$ is the *mean loss probability*, i.e., the parameter of the exponential model used in the FEC assignment exponential model, and $\mathrm{clp} = 1 - q$ is the *conditional loss probability*, conditioned on the event that the previous packet was lost.

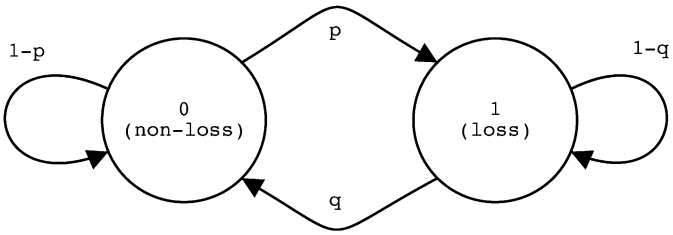


Fig. 3. Gilbert Model.

TABLE IV
Channel Loss Test Conditions

| condition | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Conditional Loss Probability (clp) | 0.147 | 0.33 | 0.50 | 0.60 |
| Mean Loss Probability (mlp) | 0.006 | 0.09 | 0.286 | 0.385 |

### B. Baselines

We compared our FEC approach to a number of alternatives. For a worst case baseline, we replace the missing frames with the mean frame over all training data.[4] A more reasonable baseline is one that uses interpolation to replace lost frames. In this case, because of the constraint of using packets efficiently, 3 frames are lost when a packet is lost and these are interpolated using a linear combination of the 2 nearest correctly received neighbors on each side, with the weights being a linear function of the distance of the missing frame from the corresponding neighbor.

We also compared the more sophisticated channel coding algorithm to a multiple transmission scheme that is a type of repetition code. In a sequence of 6 frames, the baseline system will use 2 packets, where the first packet will have frames $\{1, 2, 3\}$ and second packet will have frames $\{4, 5, 6\}$. In the multiple transmission method a sequence of 6 frames is packetized using 4 packets. Packet 1 will have frames $\{1, 2, 3\}$, packet 2 will have frames $\{2, 3, 4\}$, packet 3 will have frames $\{3, 4, 5\}$ and packet 4 will have frames $\{4, 5, 6\}$. This means that we essentially double the transmission bit rate. While this does not give a balanced transmission (frames 3 and 4 are sent 3 times, frames 1 and 6 are sent once), it does give the smallest possible latency (frame 4 is received in the second packet versus the third in a strict repetition code). The multiple transmission scheme is also combined with interpolation.

### C. Test Results

Experiments were conducted under each of the four loss conditions. Performance analyzes in Fig. 4 and Tables V and VII are based on the development set, also used to find the ULP bit assignments to subvectors. Results summarizing the performance of different source and channel coders reported in Table VI are conducted on the independent evaluation set. Recall that the baseline (no packet loss) performance for the quantized features

[4]In fact, we tried three different baselines that did not involve changes to the recognizer: replace the lost frame with zeros, drop the frame from the received sequence, and replace the frame with the training data mean. Results here are for the third case, which gave the best performance; dropping the lost frame gave the worst performance.

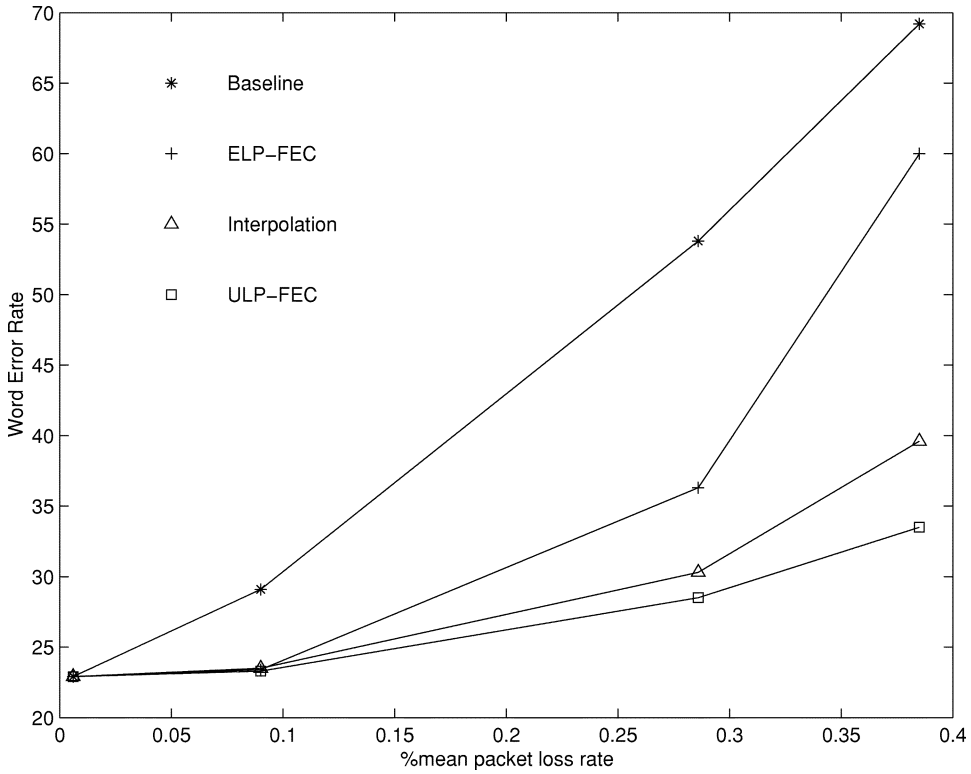


Fig. 4. Percent WER on the development set for the baseline and interpolation conditions compared to forward error correction with equal and unequal loss protection, for channel conditions with increasing mean loss probability (mlp).

TABLE V
PERCENT WER ON THE DEVELOPMENT SET FOR FORWARD ERROR CORRECTION WITH ULP USING DIFFERENT LATENCIES IN LOW, MEDIUM AND HIGH LOSS CONDITIONS. IN ALL CASES, THE TOTAL RATE IS 4.8 Kbps

| | channel mlp | | |
|---|---|---|---|
| Latency | .09 | .29 | .38 |
| 50ms | 23.7 | 33.1 | 37.7 |
| 200ms | 23.3 | 31.0 | 35.5 |

is 22.9% and 24.8% WER, respectively for the development and evaluation test sets.

Fig. 4 shows the WER results on the development set for the worst case baseline and error concealment via interpolation (both at a rate of 2.6 Kbps[5]) compared to the result using FEC with equal and unequal loss protection (both at a rate of 5.2 Kbps). In all cases, the data coding rate is the same, and the error correction involves increasing the total rate. (Fixed total rate comparisons are included later.) The results clearly demonstrate that using ULP to assign more FEC to more important cepstrum coefficients leads to substantial performance gains in high loss conditions. The ULP scheme outperforms interpolation, but it is interesting to note that interpolation works much better than the equal loss protection (ELP) case.

Table V gives the performance in WER on the development set for forward error correction with ULP using 50 ms versus 200 ms latencies in different loss conditions, in all cases at a total bit rate of 4.8 Kbps. The main conclusion from experiments with different latencies is that the added latency is only useful at the higher loss rates, which is consistent with the results in Fig. 4. It was mentioned earlier that a multiple transmission scheme (a simplistic error code) could be used to reduce the latency, but experiments show that used alone it is not effective. The error

[5]While the source code rate is 2.6 Kbps, the actual rate including the full packet is 2.67 Kbps.

TABLE VI
PERCENT WER FOR DIFFERENT LOSS RECOVERY METHODS UNDER THE HIGH LOSS CONDITION ($\mathrm{mlp} = .38$), FOR BOTH DEVELOPMENT (DEV) AND EVALUATION (EVAL) TEST SETS; TOTAL RATE FOR FEC ULP AND MT IS 5.2 Kbps

| VQ | Coding Condition | DEV | EVAL |
|---|---|---|---|
| intra | Baseline | 69.2 | 69.9 |
| intra | Interpolation | 39.6 | 42.1 |
| intra | FEC ULP | 28.5 | 31.3 |
| intra | MT + interpolation | 25.1 | 27.4 |
| inter | FEC ULP | 24.0 | 26.2 |

TABLE VII
PERCENT WER ON THE DEVELOPMENT SET FOR INTERPOLATION VERSUS ULP USING DIFFERENT TOTAL BIT RATES IN LOW, MEDIUM AND HIGH LOSS CONDITIONS. IN ALL ULP CASES, THE DELAY IS 200 ms AND THE SOURCE CODER IS INTER-FRAME VQ

| | | channel mlp | | |
|---|---|---|---|---|
| method | Rate | .09 | .29 | .38 |
| Interpolation | 2.6Kbps | 24.5 | 31.0 | 39.6 |
| FEC ULP | 2.6Kbps | 23.7 | 27.0 | 32.0 |
| FEC ULP | 5.2Kbps | 22.9 | 23.5 | 24.0 |

rate under high loss conditions ($\mathrm{mlp} = .38$) is 45%, which is (not surprisingly) worse than the low-latency Reed-Solomon code result.

Both Fig. 4 and Table V show that the ULP-FEC is most effective at the highest loss rates (28.6% and 38%). While such rates are not common, high loss rates do occur at times. Studies on packet loss over the public Internet, MBone, and wireless networks verify that packet loss rates vary over time and tend to be bursty in nature. As an example, in [37] MPEG-compressed video was sent over the Internet, and average loss rates ranging from 3.0% to 13.5% were reported. In [38], it was reported that Internet links occasionally experience loss rates up to 47% (and 68% for acknowledgment packets). In [39], loss rates ranging from 5.15% to 16.98% were reported for a particular audio source broadcasting over the MBone. For wireless links, in [40] packet loss rates of 25.6% were reported when packets are sent from a mobile host to a router, and 3.6% when packets are sent from a router to a mobile host in the MosquitoNet wireless environment.

The results above, with the exception of the interpolation baseline, do not take advantage of the high degree of temporal correlation in speech. Two methods for doing so are the inter-frame VQ source coding strategy and error concealment via interpolation. Table VI gives the WER results on both test sets for the previous best case and two different methods for using temporal correlation, in this case using a 5.2 Kbps rate. Results are reported for the high loss condition only, where the differences are statistically significant for the best case schemes. We do get good performance from the low latency approach of combining multiple transmission with interpolation, but the best results are achieved using ULP with inter-frame VQ. Note also that the multiple transmission approach is not flexible in terms of bit rate, unlike the FEC approach. Our experiments combining ULP with interpolation did not give any gains in performance, mainly because the losses from which ULP cannot recover are precisely those high loss cases where interpolation is not effective.

In order to more directly compare with the baselines operating at 2.6 Kbps, and to evaluate performance with a smaller fraction of error protection bits, we implemented the best case system (inter-frame VQ with ULP) at this lower rate. Results are given in Table VII, together with the interpolation results and the higher rate coding results for reference. While it is the case that the higher rate gives a much more robust system, there is still a significant improvement in performance due to using error protection versus concealment when the rates are equivalent.

## VII. Conclusions and Future Work

In this paper, a ULP scheme for reducing the effects of packet loss was specifically tailored for ASR applications. A key difference with respect to previous work in ASR is the focus on packet loss, i.e., erasures rather than errors. A key difference with respect to speech coding work is that all codes are designed with recognition in mind, i.e., so as to minimize word error rate. In experiments simulating adverse network conditions ($\mathrm{mlp} = .38$), the WER of the baseline system triples. This degradation can be reduced by 80% with no increase in bit rate by exploiting temporal redundancy in the VQ step and using WER-driven ULP. If the combined source and channel coding rate is doubled to 5.2 Kbps, the performance loss is nearly eliminated.

There are four main conclusions from the experimental results. First, we reconfirm results in [3] that we can quantize ASR feature vectors at relatively low rates (2.6 Kbps) without observing any degradation in ASR performance. We can further reduce the rate to 1.2 Kbps when using inter-frame VQ. The reduced bit rate is important, because it allows for introducing an error correction code. The second conclusion is that unequal loss protection, as opposed to equal loss protection, is critical for effective use of forward error correction for ASR. Third, we observe that the added code complexity and latency associated with Reed-Solomon coding is useful mainly in higher loss conditions. Lastly, we find that forward error correction outperforms error concealment via interpolation, but that combining interpolation with error protection does not further improve performance. However, it may still be possible to improve performance by combining error correction with error concealment if the error correction code is designed so as to account for the concealment procedures.

It is likely that the current system can be further improved by changing the quantization strategy. Bernard and Alwan [20] report a combined source and channel code rate of 1 Kbps by combining product code VQ with linear prediction, though this result is for a digit recognition task and it may be that a higher rate is required for larger vocabularies. In addition, it may be useful to alter the training of the quantizer, since the bit allocation experiments showed that reduced Euclidean distance was not a good indicator of improved WER. A more suitable criterion (but still practical) would be maximum likelihood [43].

Another direction for extending this work is relaxing the constraint that the core recognizer should not be changed. If we can change the front end, then it is possible to process full-bandwidth speech rather than telephone band (as used here), which has been shown to give performance gains [3]. Furthermore, it appears that lower rates are achievable with features other than MFCCs, as in [17], [18]. If we can change the acoustic model of the recognizer, then it can be more cost-effective to use discrete mixture models rather than continuous (Gaussian mixture) distribution models [21], taking advantage of the fact that the cepstral features are quantized and the observation probabilities can be obtained by a table lookup.

## Acknowledgment

The authors gratefully acknowledge the help and advice of S. Schwarm, A. Mohr, and P. Gavalakis, and we thank the speech groups at CU and CMU for providing Communicator data that was used in this study. We also give many thanks to Nuance for providing the recognizer used in this study.

## References

[1] X. Huang *et al.*, "Mipad: A multimodal interaction prototype," in *Proc. ICASSP*, vol. 1, 2001, pp. 9–12.

[2] R. C. Rose, S. Parthasarathy, B. Gajic, A. E. Rosenberg, and S. Narayanan, "On the implementation of ASR algorithms for hand-held wireless mobile devices," in *Proc. ICASSP*, vol. 1, 2001, pp. 17–20.

[3] V. Digalakis, L. Neumeyer, and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the World Wide Web," *IEEE J. Select. Areas Commun.*, vol. 17, pp. 82–90, Jan. 1999.

[4] S. Maes, D. Chazan, G. Cohen, and R. Hoory, "Conversational networking: Conversational protocols for transport, coding and control," in *Proc. ICSLP*, vol. II, 2000, pp. 198–201.
[5] V. Hardman, A. Sasse, M. Handley, and A. Watson, "Reliable audio for use over the Internet," in *Proc. INET'95*, June 1995, pp. 171–178.
[6] S. Euler and J. Zinke, "The influence of speech coding algorithms on automatic speech recognition," in *Proc. ICASSP*, vol. 1, 1994, pp. 621–624.
[7] B. Lilly and K. Paliwal, "Effect of speech coders on speech recognition performance," in *Proc. ICSLP*, vol. 4, 1996, pp. 2344–2347.
[8] T. Quatieri, E. Singer, R. Dunn, D. Reynolds, and J. Campbell, "Speaker and language recognition using speech coded parameters," in *Proc. Eurospeech*, vol. 2, 1999, pp. 787–790.
[9] L. Besacier, S. Grassi, A. Dufaux, M. Ansorge, and F. Pellandini, "GSM speech coding and speaker recognition," in *Proc. ICASSP*, vol. II, 2000, pp. 1085–1088.
[10] J. Huerta and R. Stern, "Speech recognition from GSM codec parameters," in *Proc. ICSLP*, vol. 4, 1998, pp. 1463–1466.
[11] A. Gallardo, F. Diaz, and F. Vavlerde, "Avoiding distortions due to speech coding and transmission errors in GSM ASR tasks," in *Proc. ICASSP*, vol. 1, May 1999, pp. 277–280.
[12] C. Peláez-Moreno, A. Gallardo-Antonlín, and F. Díaz de María, "Recognizing voice over IP: A robust front-end for speech recognition on the World Wide Web," *IEEE Trans. Multimedia*, vol. 3, pp. 209–218, June 2001.
[13] T. Quatieri, R. Dunn, D. Reynolds, J. Campbell, and E. Singer, "Speaker recognition using G.729 speech codec parameters," in *Proc. ICASSP*, vol. II, 2000, pp. 1089–1092.
[14] H. K. Kim and R. Cox, "A bitstream-based feature extraction for wireless speech recognition on IS-136 communications system," *IEEE Trans. Speech Processing*, vol. 9, no. 5, pp. 558–568, 2001.
[15] D. Chazan, G. Cohen, R. Hoory, and M. Zibulski, "Low bit rate speech compression for playback in speech recognition systems," in *Proc. Eur. Signal Processing Conf.*, 2000.
[16] G. N. Ramaswamy and P. S. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments," in *Proc. ICASSP*, vol. 2, 1998, pp. 977–980.
[17] Q. Zhu and A. Alwan, "An efficient and scalable 2D DCT-based feature coding scheme for remote speech recognition," in *Proc. ICASSP*, vol. I, May 2001, pp. 113–116.
[18] A. Bernard and A. Alwan, "Source and channel coding for remote speech recognitions over error-prone channels," in *Proc. ICASSP*, vol. IV, May 2001, pp. 2613–2616.
[19] A. Potamianos and V. Weerackody, "Soft-feature decoding for speech recognition over wireless channels," in *Proc. ICASSP*, vol. I, May 2001, pp. 269–272.
[20] A. Bernard and A. Alwan, "Joint channel decoding—Viterbi recognition for wireless applications," in *Proc. Eurospeech*, vol. 4, Sept. 2001, pp. 2704–2706.
[21] V. Digalakis, S. Tsakalidis, C. Harizakis, and L. Neumeyer, "Efficient speech recognition using subvector quantization and discrete-mixture HMMs," *Comput. Speech Lang.*, vol. 14, no. 1, pp. 33–46, Jan. 2000.
[22] N. S. Jayant and S. W. Christensen, "Effects of packet loss in waveform coded speech and improvements due to odd-even sample-interpolation procedure," *IEEE Trans. Commun.*, vol. 29, pp. 101–109, Feb. 1981.
[23] J.-C. Bolot, S. Fosse-Parisis, and D. Towsley, "Adaptive FEC-based error control coding for Internet telephony," in *Proc. INFOCOM*, vol. 3, 1999, pp. 1453–1460.
[24] R. Arean, J. K. Kovačević, and V. K. Goyal, "Multiple description perceptual audio coding with correlating transforms," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 140–145, Mar. 2000.
[25] W. Jiang and A. Ortega, "Multiple description speech coding for robust communication over lossy packet networks," in *Proc. ICME*, vol. 1, 2000, pp. 444–447.
[26] J.-F. Wang, J.-C. Wang, J.-F. Yang, and J.-J. Wang, "A voicing-driven packet loss recovery algorithm for analysis-by-synthesis predictive speech coders over Internet," *IEEE Trans. Multimedia*, vol. 3, pp. 98–107, Mar. 2001.
[27] B. Milner and S. Semnani, "Robust speech recognition over IP networks," in *Proc. ICASSP*, vol. III, June 2000, pp. 1791–1794.
[28] B. Milner, "Robust speech recognition in burst-like packet loss," in *Proc. ICASSP*, vol. I, May 2001, pp. 261–264.
[29] B. Pellom, W. Ward, J. Hansen, K. Hacioglu, J. Zhang, X. Yu, and S. Pradhan, "University of Colorado dialog systems for travel and navigation," in *Proc. Human Language Technology Conf.*, Mar. 2001.
[30] M. Eskenazi, A. Rudnicky, K. Gregory, P. Constantinides, R. Brennan, C. Bennett, and J. Allen, "Data collection and processing in the Carnegie Mellon Communicator," in *Proc. Eurospeech*, vol. 6, 1999, pp. 2695–2698.
[31] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*. Boston, MA: Kluwer, 1992.
[32] M. J. Sabin and R. M. Gray, "Product code vector quantizers for waveform and voice coding," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, pp. 474–488, June 1984.
[33] A. Buzo, A. H. Gray Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 28, pp. 562–574, Oct. 1980.
[34] P. A. Chou, T. Lookabaugh, and R. M. Gray, "Optimal pruning with applications to tree-structured source coding and modeling," *IEEE Trans. Inform. Theory*, vol. 35, pp. 299–315, Mar. 1989.
[35] L. Rizzo, "Effective erasure codes for reliable computer communication protocols," *ACM Comput. Commun. Rev.*, vol. 27, no. 2, pp. 24–36, Apr. 1997.
[36] A. E. Mohr, E. A. Riskin, and R. E. Ladner, "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction," *IEEE J. Select. Areas Commun.*, vol. 18, no. 6, pp. 819–828, June 2000.
[37] J. M. Boyce and R. D. Gaglianello, "Packet loss effects on MPEG video sent over the public Internet," *ACM Multimedia*, pp. 181–190, 1998.
[38] V. Paxson, "End-to-end Internet dynamics," *IEEE/ACM J. Networking*, vol. 7, no. 3, pp. 277–292, June 1999.
[39] M. Yajnik, J. Kurose, and D. Towsley, "Packet loss correlation in MBone multicast network," in *Proc. IEEE Global Internet Miniconference, Part of GLOBECOM'96*, Nov. 1996, pp. 94–99.
[40] K. Lai, M. Roussopoulos, D. Tang, X. Zhao, and M. Baker, "Experiences with a mobile testbed," in *Proc. 2nd Int. Conf. Worldwide Computing and Its Applications*, Mar. 1998.
[41] W. Jiang and H. Schulzrinne, "Modeling of packet loss and delay and their effect on real-time multimedia service quality," in *Proc. 10th Int. Workshop Network and Operations System Support for Digital Audio and Video*, June 2000.
[42] H. Sanneck and N.-T. L. Le, "Speech property-based FEC for Internet telephony applications," in *Proc. SPIE/ACM SIGMM Conf. 2000*, Jan. 2000, pp. 38–51.
[43] M. Ostendorf and J. R. Rohlicek, "Joint quantizer design and parameter estimation for discrete hidden Markov models," in *Proc. ICASSP*, vol. 2, Apr. 1990, pp. 705–708.

**Constantinos Boulis** received the M.S. degree in 2000 in computer engineering from the Technical University of Crete, Greece, from where he also holds an undergraduate degree. He is pursuing the Ph.D. degree in electrical engineering at the University of Washington, Seattle.

His academic interests include unsupervised topic detection in unconstrained speech, distributed speech recognition, speaker adaptation, and pattern recognition in general.

**Mari Ostendorf** (M'85–SM'97) received the B.S., M.S., and Ph.D. degrees in 1980, 1981, and 1985, respectively, all in electrical engineering from Stanford University, Stanford, CA.

In 1985, she joined the Speech Signal Processing Group at BBN Laboratories, where she worked on low-rate coding and acoustic modeling for continuous speech recognition. She joined the faculty of the Department of Electrical and Computer Engineering at Boston University in 1987, and since 1999 she has been a Professor of electrical engineering at the University of Washington. Her research interests are primarily in the area of statistical pattern recognition for nonstationary processes, particularly in speech processing applications, and her work has resulted in more than 100 publications. She has made contributions in segment-based and higher-order acoustic models, data selection and transformation for language modeling, and stochastic models of prosody for both recognition and synthesis.

Dr. Ostendorf has served on the Speech Processing and DSP Education Committees of the IEEE Signal Processing Society and is a member of Sigma Xi.

**Eve A. Riskin** (M'90–SM'99) received the B.S. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1984, and the M.S. degree in electrical engineering in 1985, the M.S. degree in operations research in 1986, and the Ph.D. degree in electrical engineering in 1990, all from Stanford University, Stanford, CA.

Since September 1990, she has been at the University of Washington where she is now Professor of electrical engineering and Director of the ADVANCE Center for Institutional Change. Her research interests include image compression and image processing.

Dr. Riskin was awarded the National Science Foundation Young Investigator Award in 1992 and the Sloan Research Fellowship in 1994.

**Scott Otterson** received the B.S.E.E. degree from the University of Minnesota, Minneapolis, in 1984 and the M.S.E.E. degree from Marquette University, Milwaukee, WI, 1992. He is currently a Ph.D. student in electrical engineering at the University of Washington, Seattle.

He has been a Medical Imaging Systems Engineer at General Electric Medical Systems and also at Siemens Medical Systems. He has also designed cellular RF pattern recognition systems at Cellular Technical Services, Inc. His academic interests include speaker segmentation in multiparty meetings and distributed speech recognition.