A Bitstream-Based Front-End for Wireless Speech Recognition on IS-136 Communications System

Hong Kook Kim, *Senior Member, IEEE,* and Richard V. Cox, *Fellow, IEEE*

***Abstract*—In this paper, we propose a feature extraction method for a speech recognizer that operates in digital communication networks. The feature parameters are basically extracted by converting the quantized spectral information of a speech coder into a cepstrum. We also include the voiced/unvoiced information obtained from the bitstream of the speech coder in the recognition feature set. We performed speaker-independent connected digit HMM recognition experiments under clean, background noise, and channel impairment conditions. From these results, we found that the speech recognition system employing the proposed bitstream-based front-end gives superior word and string accuracies over a recognizer constructed from decoded speech signals. Its performance is comparable to that of a wireline recognition system that uses the cepstrum as a feature set. Next, we extended the evaluation of the proposed bitstream-based front-end to large vocabulary speech recognition with a name database. The recognition results proved that the proposed bitstream-based front-end also gives a comparable performance to the conventional wireline front-end.**

***Index Terms*—Bitstream-based front-end, feature extraction, frame erasure concealment, speech coding, speech enhancement, wireless speech recognition.**

I. Introduction

THERE has been much effort to realize speech recognition technology in digital communication networks. The research work, especially on a front-end design for wireless speech recognition, can be classified into three categories as shown in Fig. 1. The synthesized speech from a speech coder is used as an input to an automatic speech recognizer (ASR) as shown in Fig. 1(a). In this case, the recognition accuracy is significantly degraded compared to that of the corresponding wireline ASR, where *wireline* means that an ASR is trained and tested with speech signals before being encoded by a speech coder. In order to improve recognition performance, some adaptation and/or normalization techniques particular to the wireless environment should be applied to the ASR system. Research in this field was reported in the global system for mobile communication (GSM) environment [1]–[4].

A second approach, shown in Fig. 1(b), is to develop a coder that quantizes and transmits speech recognition parameters. In other words, the coder performs a speech analysis for speech recognition not speech coding. This scheme has many applications for Internet access. When the mel-frequency cepstral coefficients are compressed at a rate of around 4 kbit/s, the ASR at the decoder side of the coder has performance comparable to the conventional wireline ASR system [5], [6]. However, this scheme is not able to generate synthesized speech of high quality if the detailed excitation information is not transmitted to the speech decoder. On the other hand, the first approach generates the speech signal with the quality of the speech coder. We aim to develop an ASR front-end whose performance is comparable to wireline ASR and is able to keep decoded speech of high quality. Therefore, we considered the following approach.

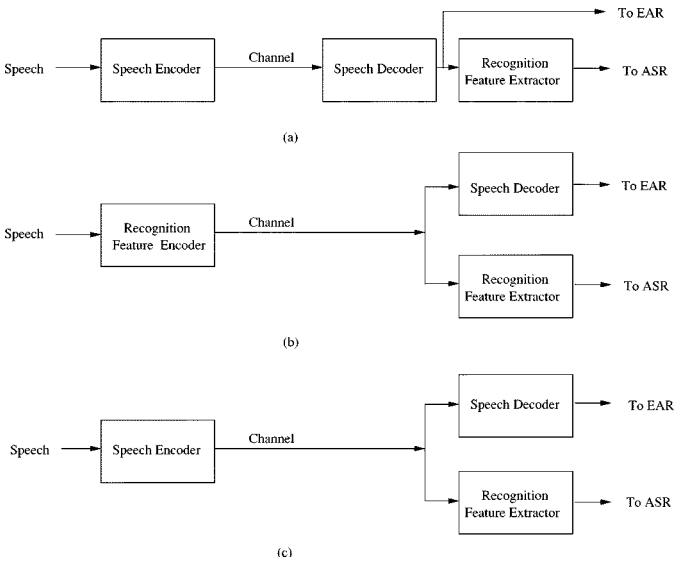

Fig. 1. Classification of wireless automatic speech recognition systems: (a) a conventional approach using a feature set extracted from decoded speech, (b) a recognition feature compression scheme, and (c) the proposed approach using a feature set extracted from a bitstream of the speech coder.

As shown in Fig. 1(c), we obtain speech recognition parameters from the bitstream of a speech coder. This is not new. Similar work has been reported as follows: The regular-pulse excitation long-term prediction (RPE-LTP) speech coder [7] has been adopted as a standard speech coder for the GSM system. The speech coder quantizes log area ratios (LARs) converted from linear predictive coding (LPC) coefficients of order eight. In [8], the authors reported that for isolated digit recognition the feature parameters derived from eight LARs and the log energy of the decoded speech do not decrease recognition performance significantly. Also, the authors of [9] used quantized residual cepstra as well as LPC cepstra for decoded speech recognition and obtained a comparable recognition performance to wireline ASR. The quantized residual signal could have some spectral information because RPE-LTP only represents spectral envelope with eight LARs, while most speech recognizers adopt $10 \sim 12$

Manuscript received February 16, 2000; revised January 25, 2001. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Hsiao-Wuen Hon.

The authors are with Shannon Laboratories, AT&T Laboratories—Research, Florham Park, NJ 07932-0971 USA (e-mail: hkkim@research.att.com; rvc@research.att.com).

Publisher Item Identifier S 1063-6676(01)04983-5.

LPC or LPC-related coefficients as a recognition feature set. Additionally, recognition experiments with the bitstream from the LPC-10e coder and Qualcomm CELP were reported in [10] and [11], respectively. Moreover, a parameter conversion for an efficient line spectrum pairs (LSP)-based speech recognition was proposed in [12], where the authors proposed the so-called pseudo-cepstral coefficients that are simply obtained from LSPs. Most of the results showed that under channel error as well as clean environments, extracting speech recognition parameters from the bitstream gives higher and more reliable recognition accuracy than obtaining the parameters from the decoded speech signal. Nevertheless, the recognition performance over low bit-rate speech coding is still worse than that of wireline ASR [13].

In this paper, we propose a new front-end based on the bitstream of a speech coder. In addition to the cepstrum directly converted from the spectral information bits of the speech coder, we incorporate voicing parameters into the elements of the feature set. A speech coder based on the speech production model decomposes the speech signal as a vocal tract filter and excitation signal. The excitation signal is separated into voiced and unvoiced components. The pitch period and its periodic correlation, which is the so-called long-term prediction or adaptive codebook gain, are used to model a voiced component. The unvoiced component is represented as random or well-structured excitation multiplied by a proper gain value, which is called the fixed codebook gain. Therefore, the adaptive codebook gain and the fixed codebook gain are able to represent the voiced/unvoiced information of speech and act as soft decision information of a voiced/unvoiced classification. Throughout this work, we choose the IS-136 digital cellular system as a wireless communications system and the IS-641 speech coder as a speech coder [14]. Since adverse conditions such as background noise and frame erasures, which degrade speech recognition performance, often exist in wireless communications networks, recognition performance of the proposed front-end should be reliable under such conditions. We can also apply our proposed method to other speech coders that operate in wireless communications networks because the current standard speech coders for wireless communications are based on the same structure as IS-641, which is called code-excited linear prediction (CELP) [15].

This paper is divided into four parts. The first part is composed of Sections II and III in which we briefly describe the conventional front-end for wireline ASR and review the speech analysis of the IS-641 coder in Sections II-A and II-B, respectively. Next, the procedure used for deriving cepstral coefficients from the bitstream of the speech coder is described in Section II-C. In Section III, we compare the recognition performance of wireless ASR employing the baseline bitstream-based front-end to those of wireline and wireless ASRs with the conventional front-end. The second part, given in Sections IV and V, proposes an improved bitstream-based front-end and verifies the performance improvement. In other words, the bitstream-based front-end is further improved by incorporating adaptive codebook and fixed codebook gains into a feature set. We introduce the matched-pairs test and McNemar's test to show how the word and the string accuracies are improved significantly compared to the conventional approach. The third part of the paper, given in Section VI, presents speech recognition experiments with two other databases: the TIDIGITS and a large vocabulary name database. We compare the word recognition accuracies of the conventional wireline and the bitstream-based front-ends. The aim of this section is to show the effectiveness of the proposed bitstream-based front-end to other database. The fourth part of the paper presents the performance of the proposed front-end in adverse communications environments that include acoustic background noise in Section VII as well as channel impairments in Section VIII. We apply a speech enhancement technique to improve recognition performance under noise conditions and also propose two processing techniques under a frame erasure channel. Finally, we present our conclusions in Section IX.

II. Feature Extraction

This section is composed of three parts. First, we review a conventional front-end used in this work. Second, we briefly explain the IS-641 speech coding algorithm in a view of speech analysis and describe the difference of spectral analysis in between the speech coder and the conventional front-end. Third, we describe the bitstream-based front-end that extracts cepstral coefficients from the bitstream of the speech coder.

A. Conventional Feature Extraction

In a conventional ASR front-end, the speech signal is preemphasized by using a first order differentiator of $1-0.95z^{-1}$. Next, the speech signal is time-limited with a window in order to obtain a linear prediction polynomial by using the autocorrelation method. The type and the length of the window are determined as a tradeoff between time and frequency resolutions. Typically, a Hamming window of length 30 ms is applied to the speech segment. The Levinson-Durbin recursion is applied to the autocorrelation coefficients to extract LPC coefficients of order ten. Finally, the LPC-derived cepstral coefficients are computed up to the 12 order, and then a cepstral lifter can be applied to the cepstral coefficients [16]. This analysis is repeated once every 10 ms, which results in a frame rate of 100 Hz. Fig. 2(a) shows this procedure for conventional feature extraction.

B. Review of Spectral Analysis in Speech Coding

Fig. 2(b) shows the simplified block diagram of the LPC analysis of the IS-641 speech coder. The speech coder removes undesired low frequency components from the speech signal by application of a highpass filter whose cutoff frequency is 80 Hz. The speech coder uses an asymmetric window, where one side of the window is a half of Hamming window and the other is a quarter period of the cosine function. This strange shape is due to the limited lookahead of the speech coder to minimize delay for real applications. Two processes are additionally applied to the autocorrelation sequence. One is lag-windowing, and the other is white noise correction. The former helps smooth the LPC spectrum and have no sharp spectral peaks [17]. The latter gives the effect of adding white noise to the speech signal and thus avoids modeling the antialiasing filter response at high frequencies with the LPC coefficients [18]. Finally, the conventional LPC recursion is performed with this modified autocorrelation sequence. LPC coefficients of order ten are converted

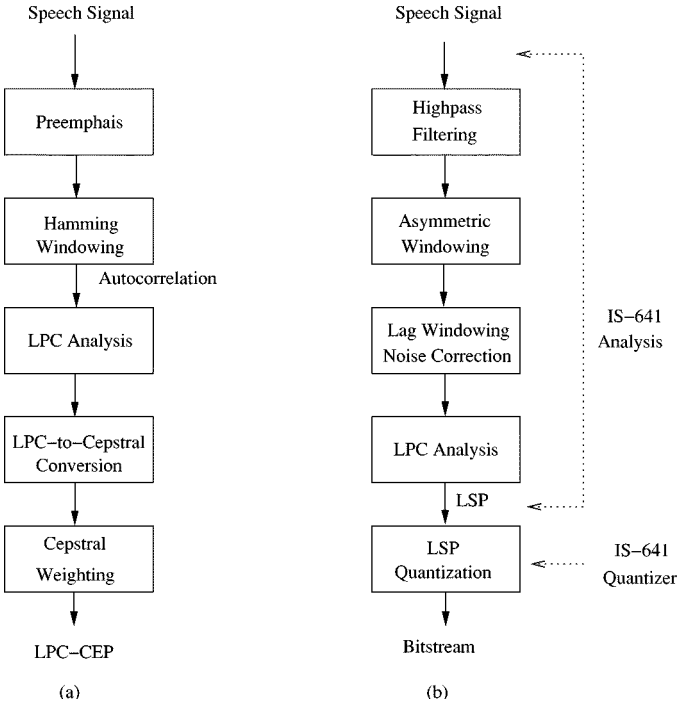

Fig. 2. Feature extraction procedures of a speech recognizer using (a) direct input speech and (b) the IS-641 speech coding parameters.

into ten LSPs. The speech encoder quantizes the LSPs and then transmits them to the decoder. When we recover the LSPs at the decoder side of the speech coder, the decoded LSPs are somewhat different from the unquantized LSPs depending on the performance of the spectral quantizer of LSPs.

As described in the above, the two main differences between the spectral analysis in the conventional feature extraction and that in the IS-641 speech coding are the effects of LSP quantization and the frame rate of 50 Hz, rather than the usual 100 Hz.

C. Baseline of Bitstream-Based Feature Extraction

Fig. 3 shows the procedure for extracting cepstral coefficients from the bitstream of the IS-641 speech coder. The bitstream for a frame is largely divided into two classes. One is 26 bits for LSP quantization and the other is 122 bits for residual information. We first obtain the decoded LSPs of order ten from the 26 bits, where these LSPs represent spectral envelope of a 30-ms speech segment with a frame rate of 50 Hz. In order to match the frame rate with that of the conventional front-end explained in Section II, we interpolate these LSPs with those of the previous frame and, thus, convert the frame rate to 100 Hz. Next, cepstral coefficients of order 12 are obtained from the conversion of LSP to LPC followed by LPC-CEP conversion. We obtain the 12 liftered cepstral coefficients by applying the bandpass lifter to the cepstral coefficients. Also, we decode the residual signal from the adaptive codebook, the shape or algebraic codebook, and the codebook gains, then, we compute an energy parameter by taking the logarithm to the square-sum of the residual of 20 ms long.

III. Recognition Experiments and Discussion

In this section, we performed connected digit recognition experiments to evaluate the performance of a baseline wireless front-end, and compared it with the performance of the conventional wireline front-end. The HMM structure and subword models are the same as those of [19]. Each digit is modeled by a set of left-to-right continuous density HMMs. In this task, we used a total of 274 context-dependent subword models, which were trained by maximum likelihood estimation. Subword models contain a head-body-tail structure. The head and tail models are represented with three states, and the body models are represented with four states. Each state has eight Gaussian mixtures. Silence is modeled by a single state with 32 Gaussian mixtures. As a result, the recognition system has 274 subword HMM models, 831 states, and 6672 mixtures since there are 11 body models, 131 head models, 131 tail models, and one silence model. The training set and the testing set consist of 9766 and 1584 digit strings, respectively, which are collected in a telephone environment. The length of all the digit strings for the training varies randomly but the length for all the testing digit strings is 14. The recognition experiments are done with an unknown length grammar.

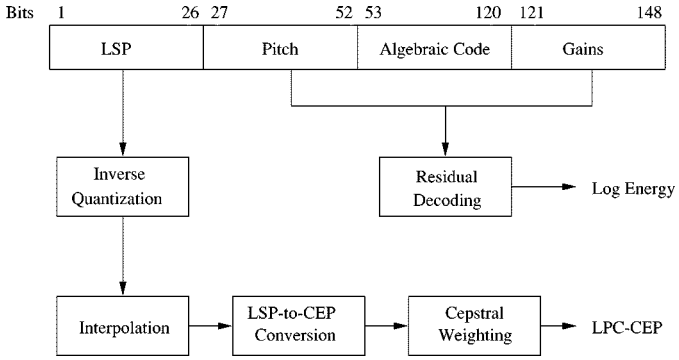

Fig. 3. Procedure of extracting feature parameters from the bitstream ofthe speech coder, where the speech coder transmits 148 bits per 20 ms.

Spectral analysis for the conventional feature extraction is done as described in Section II-A. For all training and testing strings, a logarithmic energy normalization and the cepstral mean subtraction (CMS) [19] are applied. The feature vector is 39-dimensional including 12 LPC-cepstral coefficients, a normalized logarithmic energy, and their first and second time differences. The cepstral coefficients are postprocessed by CMS, and the first and the second differences are computed from five and three frame windows, respectively.

Fig. 4 shows the configuration of each recognition system. $C0$ corresponds to the conventional wireline recognition system. In $C1$, ASR is performed on a speech signal coded by IS-641 as shown in Fig. 1(a). In order to show the recognition performance of the LPC analysis method of the speech coder, we can construct an ASR at $C2$ using the unquantized LSPs. Contrary to the conventional approach of an ASR in $C1$, we can construct an ASR at $C3$ by directly converting the bitstream of the IS-641 coder into the speech recognition feature set. In other words, $C3$ corresponds to the ASR employing the proposed bitstream-based front-end in Section II-C.

Tables I and II show the recognition accuracies for each ASR pair, where Cx/Cy means that an ASR is trained in Cx and then tested in Cy. From the results, we have the following remarks.

- We first evaluated the performance of the wireline ASR and showed the result in the second row of Table I. How-

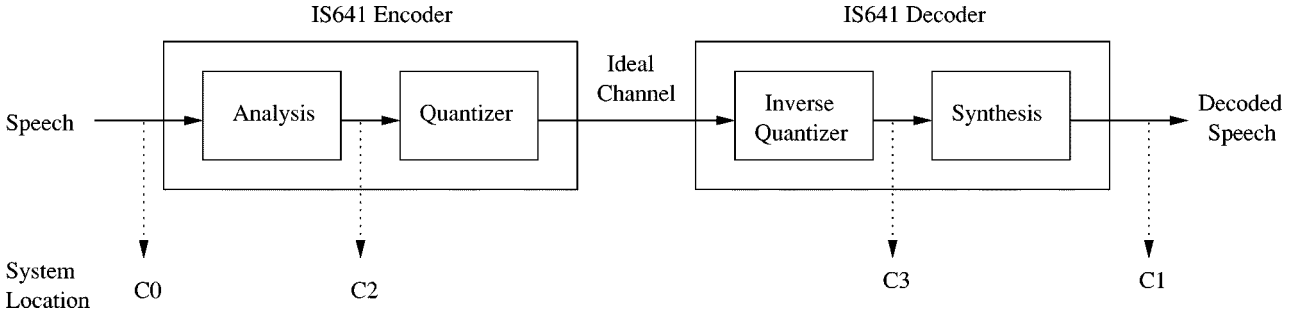

Fig. 4. Feasible locations of a speech recognizer with a combination of a speech coder: $C0$ and $C1$ mean the conventional wireline and wireless speech recognition systems, respectively; $C2$ uses the LPC analysis of IS-641 encoder before quantization; and $C3$ means bitstream-based wireless speech recognition.

TABLE I
RECOGNITION ACCURACY OF EACH CONVENTIONAL FRONT-END

Feature	Word	Word Error (%)			String
	Accuracy (%)	Sub.	Del.	Ins.	Accuracy (%)
C0/C0 (10 ms)	96.17	1.46	0.78	1.59	68.5
C0/C0 (20 ms)	95.81	1.60	0.76	1.83	66.1
C0/C1	95.16	2.09	0.95	1.79	62.3
C1/C1	94.75	2.38	1.01	1.86	60.2

TABLE II
RECOGNITION ACCURACY OF BITSTREAM-BASED FRONT-ENDS CONSTRUCTED IN THE ENCODER SIDE AND THE DECODER SIDE OF THE SPEECH CODER

Feature	Word	Word Error (%)			String
	Accuracy (%)	Sub.	Del.	Ins.	Accuracy (%)
C2/C2	96.23	1.43	0.71	1.63	68.9
C3/C3	95.81	1.68	0.82	1.69	66.5

ever, the analysis frame rate of the wireline ASR is twice that of LPC analysis of the speech coder. For a comparison of the recognition performance depending on the frame rate, we applied LPC analysis once every 20 ms to speech signals, and then obtained 10 ms feature vectors by linear interpolation. In this case, we interpolated feature vectors in the cepstral domain not in the LSP domain for simplicity. From speech coding, we know there is little spectral difference for these two domains, so we expect that there would be little difference in recognition accuracy. The word and string error rates were increased by 9% and 8%, respectively. (See second and third rows).

- $C0/C1$ means the mismatched condition between training and testing. Compared with $C0/C0$, the word and string error rates are increased by 26% and 20%, respectively. (See second and fourth rows).
- Next, we trained and tested an ASR by using the feature vectors extracted from decoded speech signals, which is denoted as $C1/C1$. Compared with the result above, higher recognition performance was expected. However, we could not improve recognition performance compared with $C0/C0$ (10 ms). The word and string errors are increased by 37% and 26%, respectively. In order to investigate why the mismatched condition provided higher performance than the matched condition, we constructed two Gaussian mixture models (GMM), λ_{C0} and λ_{C1}, by using the feature vectors from $C0$ and $C1$, respectively. Each GMM was trained by varying the number of mixtures from four to 128, and then the probability of a model preference was evaluated by counting the event that the model has the maximum *a posteriori* probability for a given observation. This procedure was similar to the work about speaker identification reported in [20]. We compared $P(X_{C0}|\lambda_{C0})$ and $P(X_{C1}|\lambda_{C0})$, where X_{C0} and X_{C1} mean the observations obtained from the testing points $C0$ and $C1$, respectively. Also, these probabilities represent the matched and mismatched conditions. As a result, $P(X_{C0}|\lambda_{C0}) > P(X_{C1}|\lambda_{C0})$ when the number of mixtures is below 16, otherwise $P(X_{C0}|\lambda_{C0}) < P(X_{C1}|\lambda_{C0})$. This explains that the recognition performance of the mismatched condition can be higher than that of the matched condition because we have trained all the HMMs for $C0$ and $C1$ with eight Gaussian mixtures except for the silence model. Also, if all the HMMs are trained with larger number of mixtures, the recognition performance of the matched condition could be better than that of the mismatched condition. (See second, fourth, and fifth rows.)
- In order to investigate the effect of the spectral quantizer in the speech coder, the recognition experiment is performed with LPC-cepstra converted from unquantized LSP. We extracted feature vectors in the IS-641 encoder before quantizing the LSP vectors and residual signal. (See second row of Table II). Surprisingly, the recognition rates are slightly higher or comparable to that of $C0/C0$. Compared with the performance of $C0/C0$ with linear interpolation, $C2/C2$ gives improvement of 10% and 8% for word and string error rates, respectively. This means that the LPC analysis of IS-641 works well for speech recognition.
- Finally, we obtained recognition rates for the proposed bitstream-based front-end. Compared with $C2/C2$, the word and string error rates of $C3/C3$ are increased by 11% and 8%, respectively. However, these results are comparable to $C0/C0$ with linear interpolation. The degradation is caused mainly by the LSP quantization in the IS-641 speech coder.

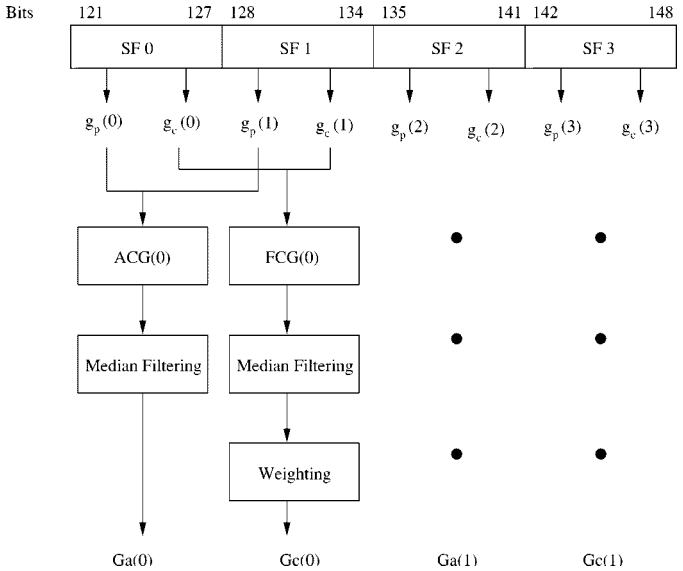

Fig. 5. Procedure of obtaining additional parameters for the proposed bitstream-based front-end.

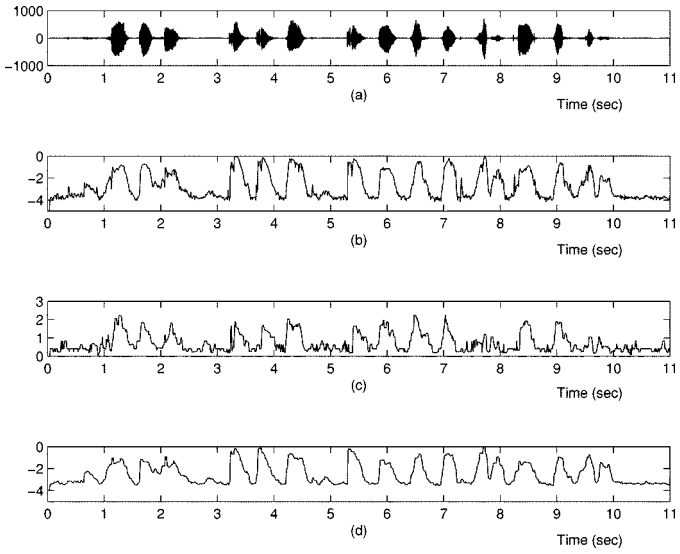

Fig. 6. Example of adaptive codebook and fixed codebook gains against speech frame for (a) a digit string; (b) normalized energy parameter, (c) adaptive codebook gain parameter, and (d) fixed codebook gain parameter.

So, we needed to devise a technique to compensate for the LSP quantization effects. One possible solution was to reestimate the quantized LSPs by using an equalization technique. Another was to incorporate other information in the feature set so that as a whole the feature set can compensate for the quantization effect.

IV. Incorporating Voiced/Unvoiced Information

In wireline ASR, some research results have been reported that have shown to improve recognition performance by using voiced/unvoiced information [21], [22]. As described in Section I, in addition to spectral envelope, the speech coder models the excitation signal as the indices and gains of the adaptive and fixed codebooks. The two gains represent the voiced/unvoiced information. These parameters are quantized and then transmitted to the decoder. So, we can implicitly obtain the voiced/unvoiced information from the bitstream. Fig. 5 shows the procedure of extracting additional parameters for the proposed bitstream-based front-end. A speech segment is further divided into four subframes and the adaptive codebook gain (ACG) and fixed codebook gain (FCG) are computed every subframe. In other words, we can get four ACGs and four FCGs for every frame. In order to make recognition feature parameters from these gains, we compute the following equations:

$$\mathrm{ACG}(i) = \sum_{k=0}^{1} g_p^2(2i+k), i = 0, 1, \quad (1)$$

$$\mathrm{FCG}(i) = \gamma \cdot 10 \log_{10} \left\{ \sum_{k=0}^{1} g_c^2(2i+k) \right\}, i = 0, 1 \quad (2)$$

where $g_p(i)$ and $g_c(i)$ are the ACG and FCG of the ith subframe. Also, in order to add ACG and FCG into a feature vector, we obtain ten LPC-cepstra instead of 12 LPC-cepstra in the baseline. Thus, the dimension of the resulting vector remains the same as the baseline front-end. Fig. 6 shows an example of the trajectories of the adaptive codebook gain and fixed codebook gain after the speech waveform is processed by the IS-641 speech coder. As can be seen, the ACG and the FCG have temporal fluctuations. To reduce the fluctuation, we apply a smoothing technique such as median filtering with five-taps to ACG and FCG. Similar to the energy parameter, we can give a weight to FCG, which controls the effect of FCG parameter relative to the others. We set γ in (2) to be 0.1.

TABLE III
Comparison of Recognition Accuracies of the Bitstream-Based Front-Ends, Where *Baseline* Corresponds to $C3/C3$ in Table II

Feature	Word	Word Error (%)			String
	Accuracy (%)	Sub.	Del.	Ins.	Accuracy (%)
C3 : Wireless Baseline	95.81	1.68	0.82	1.69	66.5
C3-1 : LPC-CEP, AFG, FCG	95.96	1.84	0.80	1.39	67.8
C3-2 : Median Smoothing	95.98	1.86	0.78	1.38	68.7
C3-3 : Gain Scaling	96.24	1.69	0.72	1.35	69.8

Table III shows the recognition results by incorporating these parameters into the recognition feature vector. Compared to the baseline, the new feature set including ACG and FCG has reduced the word and string error rates by 10% each. These results are also comparable to $C0/C0$ and $C2/C2$.

V. Analysis of Recognition Results: Significance Test

In order to analyze the recognition results, we use hypothesis tests based on the matched-pairs and McNemar's tests, for the word and string accuracy tests, respectively [23]. First, we will briefly describe the matched-pairs test.

We want to test whether the performance of a system is comparable to another or not. In other words, we can construct a hypothesis H_0 as

$$H_0 : \mu_A - \mu_B = 0 \quad (3)$$

where μ_A and μ_B represent the mean values of the recognition rates for the systems A and B, respectively. Let $R_{A,i}$ and $R_{B,i}$ be the word recognition accuracies for the ith string by A and B, respectively. Also let $Z_i = R_{A,i} - R_{B,i}$ for $i = 1, \ldots, N$, where N is the total number of strings to be tested. We can assume that Z has a normal distribution. In this case, the test statistic is given by

$$W = \frac{\hat{\mu}_Z}{\hat{\sigma}_Z/\sqrt{N}} \tag{4}$$

where $\hat{\mu}_Z = (1/N)\sum_{i=1}^{N} Z_i$ and $\hat{\sigma}_Z^2 = 1/(N-1)\sum_{i=1}^{N}(Z_i - \hat{\mu}_Z)^2$. We do not reject the hypothesis with a confidence of $(1-\alpha)$ if $W \leq z(1-(\alpha/2))$, where $z(1-(\alpha/2))$ means $\int_{-\infty}^{z} g(x)dx = 1-(\alpha/2)$ with the normal density function of $g(x)$ and $z(0.975) = 1.96$ from the mathematical table. Moreover, we can say that A is significantly superior over B if $W > z(1-(\alpha/2))$ and conversely if $W < -z(1-(\alpha/2))$.

McNemar's test can be used to test a statistical significance of two systems for string accuracy. We want to test the *Null* hypothesis: *If a string error occurs from one of the two systems, then it is equally likely to be either one of the two.* Let N_{01} be the number of strings which A recognizes correctly and B recognizes incorrectly. Similarly, we count the number of strings which A recognizes incorrectly and B recognizes correctly and denote it as N_{10}. Then the test statistics for the McNemar's test is defined by

$$W = \frac{|N_{10} - k/2| - 1/2}{\sqrt{k/4}} \tag{5}$$

where $k = N_{10} + N_{01}$. As in the matched-pairs test, W in (5) is compared with $z(1-(\alpha/2))$ for a confidence level α.

We compute the test statistic for a pair of feature sets such as wireless baseline $(C3)$ and the feature with ACG and FCG $(C3{-}3)$ and show the result in the third row of Table IV. From the table, it can be seen that the proposed feature set incorporating ACG and FCG provides significantly improved recognition performance over baseline with a confidence of 95%. Moreover, Table IV shows that the proposed front-end gives comparable word and string accuracies to conventional wireline performance.

TABLE IV
RESULTS OF MATCHED-PAIRS TEST AND MCNEMAR'S TEST FOR SEVERAL PAIRS OF FRONT-ENDS

Features		Test Result			
		Matched-pairs		McNemar	
A	B	W	Significant	W	Significant
C3-3	C3	1.965	Yes	2.445	Yes
C0	C1	3.619	Yes	3.607	Yes
C3-3	C1	3.914	Yes	4.388	Yes
C3-3	C0	0.328	No	0.833	No

TABLE V
COMPARISON OF RECOGNITION ACCURACIES OF THE THREE FRONT-ENDS ON THE TIDIGITS DATABASE: THE WIRELINE BASELINE $(C0)$, THE CONVENTIONAL WIRELESS $(C1)$, AND THE PROPOSED BITSTREAM-BASED $(C3{-}3)$ FRONT-ENDS

Feature	Word	Word Error (%)			String
	Accuracy (%)	Sub.	Del.	Ins.	Accuracy (%)
C0	98.70	0.63	0.56	0.11	96.3
C1	98.33	0.98	0.58	0.11	95.3
C3-3	99.07	0.54	0.33	0.06	97.2

VI. APPLICATION TO OTHER DATABASES

A. TIDIGITS Recognition

This subsection provides the recognition performance comparison of three front-ends denoted by $C0, C1$, and $C3{-}3$ for the TIDIGITS database [24]. As a test set, we used 8700 digit strings in the test part of the database spoken by 56 males and 57 females, where the total number of digits was 28 583. The speech utterances in the test set were down-sampled to 8 kHz. For each front-end, we used the same HMMs as described so far. This training-test pair is a kind of mismatched condition because the HMMs were trained with a database recorded in a telephone environment but the TIDIGITS database is of studio quality. Table V shows the recognition accuracies of the conventional wireline front-end $(C0)$, the conventional wireless front-end $(C1)$, and the proposed bitstream-based front-end $(C3{-}3)$. The recognition system employing the bitstream-based front-end reduced the word and string error rates by 40% and 44%, respectively, compared to that using conventional wireless front-end. Moreover, compared to the recognition system with a wireline baseline, it had a reduction in the word and string error rates by 28% and 24%, respectively. Finally, we applied the matched-pairs and McNemar's tests to the TIDIGITS test results. As a result, it was shown that the bitstream-based front-end provided a significantly better recognition accuracy than the conventional wireless front-end as well as the wireline front-end.

B. Large Vocabulary Speech Recognition

In this section, we extended the performance evaluation of the bitstream-based front-end to large-vocabulary speech recognition. The training data were 16 538 AT&T employee names that were recorded through a telephone handset. The recognition system was constructed based on context-dependent phone models and then these models were merged by using the decision tree based approach. The recognition system had 7731 phone units. For the testing, four databases were used: first-name, last-name, location, and telephone-number. Forty speakers uttered the first names and the last names of AT&T employees, and the locations where they work, and their work phone numbers in an office environment using a close-talk microphone. The sizes of each database were 2893, 2947, 2005, and 2000 for first-name, last-name, location, and telephone-number, respectively.

Fig. 7 shows the word recognition accuracy against the average processing time for each test database. By varying the beam width of a Viterbi search, we could get a different processing time on the SGI machine divided by the duration of

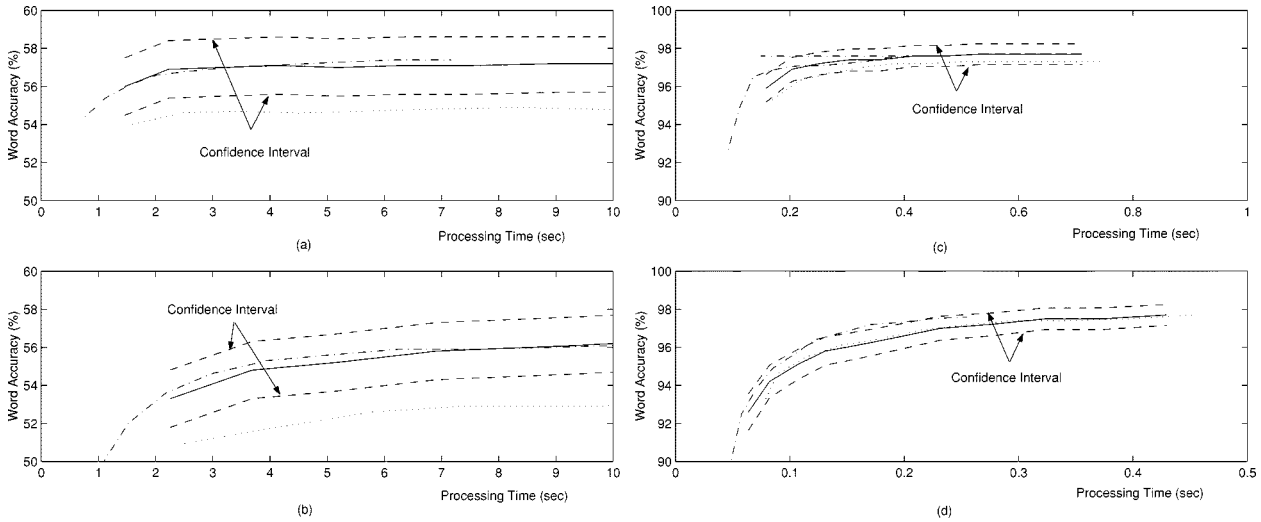

Fig. 7. Word recognition accuracy of the conventional wireline front-end (—), the decoded speech-based front-end (- - -), and the bitstream-based front-end (— · —·). Shown are the results on (a) first-name database, (b) last-name database, (c) location database, and (d) telephone-number database.

the recognized utterance.[1] In this figure, we also depicted 95% confidence intervals [25] for the conventional wireline front-end $(C0/C0)$ (as dashed lines), which indicates the statistical significance. The bitstream-based front-end (indicated as a dash dotted line) gives statistically comparable word recognition accuracies for all the test databases, and its performance is always better than that of the decoded speech-based front-end (indicated as a dotted line). These results agree with the results for connected digit recognition. The following sections show the performance of the proposed front-end in adverse communication environments such as background noise and frame erasure.

VII. Wireless Speech Recognition Under Background Noise

In this section, we investigate the performance of the proposed front-end under background noise conditions and then introduce a speech enhancement algorithm to improve the recognition performance of the front-end.

A. Background Noise

To simulate a noisy environment, a car noise signal is added to every test digit string. In other words, the speech recognition system is trained with clean speech signals and then tested with noisy signals. The amount of additive noise is controlled by the segmental signal-to-noise ratio (SNR). Table VI shows the recognition performance comparison when the input SNR varies from 0 dB to 30 dB in a step of 10 dB. For all SNRs except 0 dB, the proposed front-end always provides higher recognition rates than the conventional wireless front-end. It also gives better recognition performance at low SNRs while at 30 dB SNR its performance is slightly lower than the conventional wireline front-end.

[1]Note that recognition processing time is dependent on the size of the vocabulary. In this experiment the number of first and last names is much larger than the number of locations and telephone numbers. Actually, the vocabulary sizes for HMM decoding were 1902, 2978, 62, 11 for first names, last names, locations, and telephone numbers, respectively.

TABLE VI
Comparison of Recognition Accuracy Under the Car Noise Condition

Feature	Accuracy	SNR (dB)				
	(%)	0	10	20	30	∞
C0/	Word	14.30	61.82	85.84	95.73	96.17
C0	String	0.0	0.5	23.1	65.5	68.5
C0/	Word	21.18	65.59	85.47	94.29	95.16
C1	String	0.0	0.5	20.0	55.8	62.3
C3-3/	Word	16.82	67.28	90.64	95.28	96.24
C3-3	String	0.0	3.6	41.6	63.8	69.8

TABLE VII
Comparison of Recognition Accuracy Under the Babble Noise Condition

Feature	Accuracy	SNR (dB)				
	(%)	0	10	20	30	∞
C0/	Word	37.96	81.03	94.70	96.12	96.17
C0	String	0.0	11.9	60.9	66.3	68.5
C0/	Word	38.78	77.31	92.46	94.81	95.16
C1	String	0.0	6.8	49.1	60.3	62.3
C3-3/	Word	27.81	76.00	93.43	95.78	96.24
C3-3	String	0.0	11.6	54.3	66.1	69.8

Next, we evaluate the recognition system under babble noise condition. Table VII shows the recognition rates when the SNR varies from 0 to 30 dB in a step of 10 dB. For an SNR above 20 dB, the proposed front-end shows better performance than the conventional wireless front-end. However, its performance is slightly lower than the conventional wireline front-end. For low SNR, the proposed front-end gives the poorest performance. This is because the proposed front-end especially utilizes the voicing information but the speech coder fails to correctly capture the voicing information at low SNR. However, if we adopt

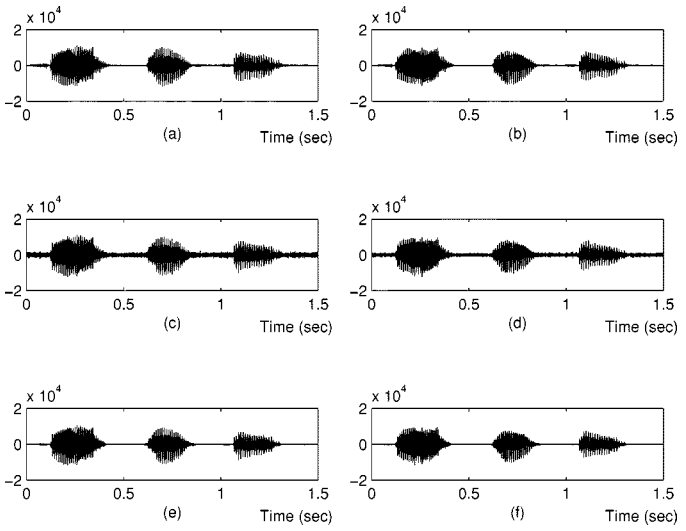

Fig. 8. Waveform examples: (a) clean speech signal whose level is −26 dBovl; (b) its decoded speech signal; (c) noisy speech signal contaminated with a car noise of 20 dB SNR; (d) its decoded speech signal; (e) enhanced signal of (c); and (f) its decoded speech signal.

a speech enhancement algorithm to the noisy speech signals before speech coding, we can extract more correct voicing information.

B. Applying Speech Enhancement Algorithm

The speech enhancement algorithm used in this work is based on minimum mean-square error log-spectral amplitude estimation and has been applied to some standard speech coders [26]–[28]. Fig. 8 displays waveforms under the various conditions. The clean speech waveform of a connected digit string is shown in Fig. 8(a). Fig. 8(b) shows the waveform decoded by the IS-641 speech coder. The noisy speech contaminated by the additive car noise, whose SNR is 20 dB, is shown in Fig. 8(c), and its decoded speech signal is displayed in Fig. 8(d). Finally, we apply the speech enhancement algorithm to the noisy speech signal [Fig. 8(e)], and then perform speech coding [Fig. 8(f)]. The figure shows that the noise signal is removed by applying the speech enhancement algorithm. Improved recognition accuracy is expected. We have done informal listening tests under the car and the babble noise conditions at 20 dB SNR. The results show that the enhancement algorithm improves the decoded speech quality and the quality improvement under the car noise is more significant than that under the babble noise. Some listeners indicated that the enhancement algorithm made some annoying sounds for the babble noisy speech.

We applied the enhancement algorithm to the noisy speech signal and then tested the speech recognition systems with the enhanced signal, where the speech enhancement algorithm was optimized in a view of speech coding not speech recognition. Table VIII shows the word and string accuracies under the car noise condition when the enhancement algorithm is applied. Compared with the results of Table VI, the enhancement algorithm reduces the recognition rates under clean conditions and high SNR. This is because the enhancement algorithm causes the spectral envelope of clean speech signal to be slightly distorted. On the other hand, the recognition performance for low SNR is greatly improved for all front-ends.

TABLE VIII
COMPARISON OF RECOGNITION ACCURACY AFTER APPLYING AN ENHANCEMENT ALGORITHM TO SPEECH SIGNAL AT THE ENCODER SIDE OF THE SPEECH CODER UNDER THE CAR NOISE CONDITION

Feature	Accuracy	SNR (dB)				
	(%)	0	10	20	30	∞
C0/	Word	53.40	88.13	94.77	95.42	95.70
C0	String	0.1	33.5	60.2	64.1	65.9
C0/	Word	55.80	87.97	93.83	94.76	94.90
C1	String	0.2	30.7	55.2	59.6	61.6
C3-3/	Word	57.47	86.61	93.93	95.60	96.03
C3-3	String	0.6	28.2	56.5	65.2	69.1

TABLE IX
COMPARISON OF RECOGNITION ACCURACY AFTER APPLYING AN ENHANCEMENT ALGORITHM TO SPEECH SIGNAL AT THE ENCODER SIDE OF THE SPEECH CODER UNDER THE BABBLE NOISE CONDITION

Feature	Accuracy	SNR (dB)				
	(%)	0	10	20	30	∞
C0/	Word	38.74	77.59	89.28	93.14	95.70
C0	String	0.0	5.2	26.2	47.2	65.9
C0/	Word	41.15	77.68	88.31	92.13	94.90
C1	String	0.0	6.1	22.6	41.2	61.6
C3-3/	Word	44.57	83.91	93.46	95.14	96.03
C3-3	String	0.0	19.6	57.8	62.0	69.1

Table IX shows the recognition results after applying the enhancement algorithm for the babble noise condition. For the conventional front-end, the enhancement algorithm reduces the recognition accuracy for 10 or 20 dB SNR, which was expected from the informal listening test. However, the recognition performance of the proposed front-end with the enhancement algorithm is comparable to or better than that without the enhancement. The proposed front-end provides the best recognition rates for all SNRs. As a result, we conclude that the proposed bitstream-based front-end with an enhancement algorithm gives comparable or better recognition rate than the conventional wireline front-end under severe noise conditions.

VIII. WIRELESS SPEECH RECOGNITION UNDER CHANNEL IMPAIRMENTS

In speech coding, channel impairments are modeled by bit error insertion and frame erasure insertion devices, where the number of bit errors and frame erasures depends mainly on the noise, co-channel and adjacent channel interface, and frequency selective fading. Fortunately, most speech coders are combined with a channel coder. The most sensitive bits are strongly protected by the channel coder. A frame erasure is declared if any of the most sensitive bits to the channel error is in error [29]. The bits for LSP and gains are classified as the most sensitive bits

to channel errors. Therefore, it is sufficient to only consider the frame erasure condition because recognition features in the bitstream-based front-end are extracted from these bits. In this section, we present two frame erasure processing techniques that are combined with the bitstream-based front-end, and then show their performance under various frame erasure conditions.

A. Use of Extrapolation Algorithm in Speech Coding

The speech coding parameters of an erased frame must be extrapolated in order to generate the speech signal for the erased frame. A family of error concealment techniques generally use substitution or extrapolation schemes [30]. The parameters of erased frames are reconstructed by repeating the parameters of the previous frame with scaled down gains. The gain values depend on the burstiness of frame erasure, where it is modeled as a finite state machine.

If the nth frame is detected as an erased frame, the IS-641 speech coder estimates the spectral parameters by using the following equation:

$$\omega_{n,i} = c\omega_{n-1,i} + (1-c)\omega_{dc,i}, i = 1, \ldots, p \tag{6}$$

where $\omega_{n,i}$ is the ith LSP of the nth frame and $\omega_{dc,i}$ is the empirical mean value of the ith LSP over a training database. c is a forgetting factor set to 0.9. Adaptive codebook and fixed codebook gains are obtained by multiplying the predefined attenuation factors to the gains of the previous frame, and the pitch value is set to the same pitch value of the previous frame. The speech signal is reconstructed from these extrapolated parameters. From now on, we will refer to this method as the *extrapolation* method. In speech recognition, a decoded speech-based front-end uses the synthesized speech for extracting a feature vector. For the bitstream-based front-end, these extrapolated parameters are directly applied to the front-end.

B. Deletion of Erased Frames

As an alternative to the extrapolation method, we propose another technique for speech recognition under an impaired channel. Based on the missing feature theory [31], [32], we reformulate a decoding algorithm for the hidden Markov model (HMM) when a frame erasure is detected. For a given HMM $\lambda = (A, B, \pi)$, the probability of the observation sequence $\mathbf{O} = \{\mathbf{o}_1, \ldots, \mathbf{o}_N\}$ is given by

$$P(\mathbf{O}|\lambda) = \sum_{q_1,\ldots,q_N} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \cdots a_{q_{N-1}q_N} b_{q_N}(\mathbf{o}_N) \tag{7}$$

where

N	number of observation vectors in $\mathbf{O}$;
$(q_1, \ldots, q_N)$	state sequence;
π_{q_1}	initial state distribution.

Also, the observation probability of $\mathbf{o}_n$ at state i is represented as

$$b_i(\mathbf{o}_n) = \sum_{k=1}^{M} c_{ik} \mathcal{N}(\mathbf{o}_n; \mu_{i,k}, \Sigma_{ik}) \tag{8}$$

where $\mathcal{N}(\mathbf{x}; \mu, \Sigma) = 1/((2\pi)^{p/2}|\Sigma|^{1/2}) \exp\{-(1/2)(\mathbf{x}-\mu)^t \Sigma^{-1}(\mathbf{x}-\mu)\}$, M is the number of Gaussian mixtures, and c_{ik} is the kth mixture weight of the ith state. Also μ and Σ are the mean vector and the covariance matrix, respectively.

Now we assume that the lth frame is detected as a missing frame. Then, we want to compute the probability of only the correct observation vector sequence for the model λ. The observation vector sequence can be divided into two groups as

$$\mathbf{O} = (\mathbf{O}^c, \mathbf{O}^m) \tag{9}$$

where $\mathbf{o}_l \in \mathbf{O}^m$. From the missing feature theory [31], [32], the probability we want to compute is

$$P(\mathbf{O}^c|\lambda) = \int P(\mathbf{O}^c, \mathbf{O}^m|\lambda) d\mathbf{O}^m. \tag{10}$$

For the missing observation vector, $\mathbf{o}_l$, we know

$$\int b_i(\mathbf{o}_l) d\mathbf{o}_l = 1. \tag{11}$$

By substituting (7) and (12) int $\mathbf{o}$ (10), we finally obtain

$$P(\mathbf{O}^c|\lambda) = \sum_{q_1,\ldots,q_N} \pi_{q_1} b_{q_1}(\mathbf{o}_1) a_{q_1 q_2} b_{q_2}(\mathbf{o}_2) \ldots a_{q_{l-1}q_l} a_{q_l q_{l+1}} b_{q_{l+1}}(\mathbf{o}_{l+1}) \ldots a_{q_{N-1}q_N} b_{q_N}(\mathbf{o}_N). \tag{12}$$

The transition probabilities have less effect in the Viterbi search than the observation probabilities [33]. Therefore, we can set $a_{q_{l-1}q_l} = 1$. Equation (12) is simply realized by deleting the $\mathbf{o}_l$ in the observation sequence and by using the standard HMM decoding procedure. This method will be referred as to the *deletion* method.

The deletion method can be interpreted in terms of a variable frame rate analysis [35]. From (6), the Euclidean distance of LSPs between the $(n-1)$th and the nth frames is given by

$$\sum_{i=1}^{p} (\omega_{n,i} - \omega_{n-1,i})^2 = (1-c)^2 \sum_{i=1}^{p} (\omega_{n-1,i} - \omega_{dc,i})^2. \tag{13}$$

If the distance of (13) is less than or equal to a predefined threshold T, the two frames are assumed to be in the steady-state region and the LSPs of the nth frame are deleted in the observation sequence. Therefore, if we let $T = (1-c)^2 \max_{[x_1,\ldots,x_p]\in\Omega} \sum_{i=1}^{p} (x_i - \omega_{dc,i})^2$, where Ω is a p-dimensional LSP vector space, all the missing frames are deleted.

In terms of computational complexity, we can say that this approach reduces the length of the observation sequence by $N(1-p_e)$, where p_e is the frame erasure rate (FER). Therefore, the computational complexities for both the feature extraction and the Viterbi decoding will be reduced by a factor of p_e, respectively.

C. Discussion

To simulate frame erasure conditions, error patterns depending on the FER and its burstiness were generated for each test string by using the program [34]. Fig. 9(a) shows the word error rate (WER) when the random FER varies from 3% to 20%. FER of 0% means the clean environment. 3% FER is typical of a TDMA channel. At 3% FER, the WERs are increased by 6.4% and 5.3% for the bitstream-based front-ends with the extrapolation method and the deletion method, respectively, while the WER is increased by 12% for the decoded speech-based front-end.

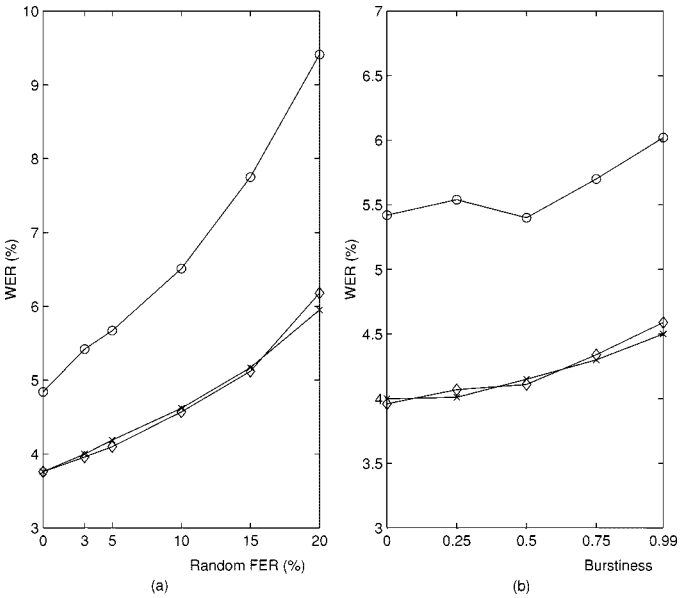

Fig. 9. Word error rate of the recognition systems employing the decoded speech-based front end ($-\ \circ\ -$), the bitstream-based front-end with the extrapolation method ($-\ \times\ -$), and the bitstream-based front-end with the deletion method ($-\ \diamond\ -$): (a) random FER and (b) burst frame erasure at the FER of 3%.

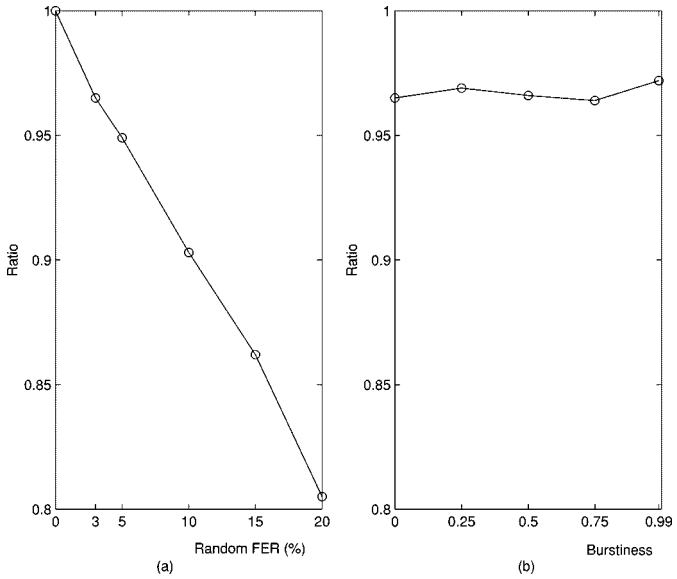

Fig. 10. Processing time ratio of the extrapolation and the deletion methods for (a) random FER and (b) burst frame erasure at the FER of 3%.

As the FER increases up to 20%, the WERs increase by about 95% and 60% for the decoded speech-based front-end and the others, respectively. For all FERs, the bitstream-based front-ends achieve lower WER than the decoded speech-based front-end. The deletion method gives a comparable WER to the extrapolation method, where the deletion method has higher deletion error and lower insertion and substitution errors than the extrapolation method.

Fig. 9(b) shows the WER according to the burstiness of FER when the FER is 3%, where the burstiness is denoted by b for the simplicity. Similar to the random FER case, the WERs of the bitstream-based front-ends are smaller than those of the decoded speech-based front-end. Comparing the WER performance at $b = 0.99$ to that under clean environment, the decoded speech-based front-end increases the WER by 24.3%, while the bitstream-based front-ends with the extrapolation method and with the deletion method increase the WER by 19.7% and 22.1%, respectively. The deletion method gives a slightly worse performance than the extrapolation method when b is large. This is because the deletion method increases the deletion errors as b increases, which is the same as the case of high random FER. Of course, we can also reduce the HMM decoding and feature extraction computation by 3%.

Fig. 10 shows the ratios of the processing time between the extrapolation and the deletion methods for each FER and burstiness. Here, the processing time was taken by performing the recognition experiments over all the test data on the SGI machine. The results verify that the proposed deletion method has less computational complexity than the extrapolation method.

IX. Conclusions

In this paper, we proposed a bitstream-based front-end for wireless automatic speech recognition. The feature parameters of the front-end consist of spectral envelope and voicing information. The spectral envelope is derived from the quantized line spectrum pairs followed by conversion to LPC-cepstral coefficients. The voiced/unvoiced information is directly obtained from the bits corresponding to adaptive and fixed codebook gains of a speech coder. From the HMM-based connected digit recognition experiments with the IS-641 speech coder, we found that a wireless ASR system employing the proposed front-end provides statistically comparable word and string accuracies to the conventional wireline ASR system using the LPC-cepstral coefficients as a feature set. In addition, we performed large vocabulary speech recognition experiments and also found that the proposed bitstream-based front-end gives a statistically comparable performance to the conventional wireline front-end. Finally, we evaluated the proposed front-end under background noise and frame erasure conditions. As a result, we found that it gives a superior performance over a conventional wireless ASR system using decoded speech signals under these conditions.

Acknowledgment

The authors would like to thank Dr. M. Rahim and Dr. R. C. Rose for helping us train the connected digit HMMs and the large vocabulary HMMs, respectively.

References

[1] S. Dufour, C. Glorion, and P. Lockwood, "Evaluation of the root-normalized front-end (RN_LFCC) for speech recognition in wireless GSM network environments," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Atlanta, GA, May 1996, pp. 77–80.

[2] T. Soulas, C. Mokbel, D. Jouvet, and J. Monne, "Adaptive PSN recognition models to the GSM environment by using spectral transformation," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Munich, Germany, Apr. 1997, pp. 1003–1006.

[3] C. Mokbel, L. Mauuary, L. Karray, D. Jouvet, J. Monne, J. Simonin, and K. Bartkova, "Toward improving ASR robustness for PSN and GSM telephone applications," *Speech Commun.*, vol. 23, no. 1–2, pp. 141–159, Oct. 1997.

[4] T. Salonidis and V. Digalakis, "Robust speech recognition for multiple topological scenarios of the GSM mobile phone system," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, May 1998, pp. 101–104.
[5] 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.
[6] G. N. Ramaswamy and P. S. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, May 1998, pp. 977–980.
[7] P. Vary, K. Hellwig, R. Hofmann, R. J. Sluyter, C. Galand, and M. Rosso, "Speech codec for the European mobile radio system," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 1, New York, Apr. 1988, pp. 227–230.
[8] A. Gallardo-Antolin, F. Diaz-de-Maria, and F. Valverde-Albacete, "Recognition from GSM digital speech," in *Proc. Int. Conf. Speech Language Processing*, Sydney, NSW, Australia, Nov. 1998, pp. 1443–1446.
[9] J. M. Huerta and R. M. Stern, "Speech recognition from GSM codec parameters," in *Proc. Int. Conf. Speech Language Processing*, Sydney, NSW, Australia, Nov. 1998, pp. 1463–1466.
[10] A.-T. Yu and H.-C. Wang, "A study on the recognition of low bit-rate encoded speech," in *Proc. Int. Conf. Speech Language Processing*, Sydney, NSW, Australia, Nov. 1998, pp. 1523–1526.
[11] S. H. Choi, H. K. Kim, H. S. Lee, and R. M. Gray, "Speech recognition method using quantised LSP parameters in CELP-type coders," *IEE Electron. Lett.*, vol. 34, no. 2, pp. 156–157, Jan. 1998.
[12] H. K. Kim, S. H. Choi, and H. S. Lee, "On approximating line spectral frequencies to LPC cepstral coefficients," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 195–199, Mar. 2000.
[13] S. H. Choi, H. K. Kim, and H. S. Lee, "Speech recognition using quantized LSP parameters and their transformations in digital communication," *Speech Commun.*, vol. 30, pp. 223–233, Apr. 2000.
[14] T. Honkanen, J. Vainio, K. Järvinen, P. Haavisto, R. Salami, C. Laflamme, and J.-P. Adoul, "Enhanced full rate speech codec for IS-136 digital cellular system," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Munich, Germany, Apr. 1997, pp. 731–734.
[15] M. Budagavi and J. D. Gibson, "Speech coding in mobile radio communications," *Proc. IEEE*, vol. 86, pp. 1402–1412, July 1998.
[16] B. H. Juang, L. R. Rabiner, and J. G. Wilpon, "On the use of bandpass liftering in speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 35, pp. 947–954, July 1987.
[17] Y. Tohkura, F. Itakura, and S. Hashimoto, "Spectral smoothing technique in PARCOR speech analysis-synthesis," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-26, pp. 587–596, Dec. 1978.
[18] B. S. Atal, "Predictive coding of speech at low bit rates," *IEEE Trans. Commun.*, vol. 30, pp. 600–614, Apr. 1980.
[19] M. Rahim, B. H. Juang, W. Chou, and E. Buhrke, "Signal conditioning techniques for robust speech recognition," *IEEE Signal Processing Lett.*, vol. 3, pp. 107–109, Apr. 1996.
[20] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using gaussian mixture speaker models," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 72–83, Jan. 1995.
[21] D. L. Thomson and R. Chengalvarayan, "Use of periodicity and jitter as speech recognition features," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Seattle, WA, May 1998, pp. 21–24.
[22] D. O'Shaughnessy and H. Tolba, "Toward a robust/fast continuous speech recognition system using a voiced-unvoiced decision," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Phoenix, AZ, Mar. 1999, pp. 413–416.
[23] L. Gillick and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Glasgow, U.K., May 1989, pp. 532–535.
[24] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 3, New York, 1984, pp. 42.11.1–42.11.4.
[25] C. E. Mokbel and G. Chollet, "Automatic word recognition in cars," *IEEE Trans. Speech Audio Processing*, vol. 3, pp. 346–356, Sept. 1995.
[26] A. J. Accardi and R. V. Cox, "A modular approach to speech enhancement with an application to speech coding," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Phoenix, AZ, Mar. 1999, pp. 201–204.
[27] D. Malah, R. V. Cox, and A. J. Accardi, "Tracking speech-presence uncertainty to improve speech enhancement in nonstationary noise environments," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Phoenix, AZ, Mar. 1999, pp. 789–792.
[28] R. Martin and R. V. Cox, "New speech enhancement techniques for low bit rate speech coding," *Proc. 1999 IEEE Workshop Speech Coding*, pp. 165–167, June 1999.
[29] N. R. Sollenberger, N. Seshadri, and R. Cox, "The evolution of IS-136 TDMA for third-generation wireless services," *IEEE Pers. Commun.*, vol. 6, pp. 8–18, June 1999.
[30] A. M. Kondoz, *Digital Speech: Coding for Low Bit Rate Communications Systems*. New York: Wiley, 1994.
[31] S. Ahmad and V. Tresp, "Some solutions to the missing feature problem in vision," in *Advances in Neural Information Processing Systems 5*, S. J. Hanson, J. D. Cowan, and C. L. Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp. 393–400.
[32] R. P. Lippmann and B. A. Carlson, "Using missing feature theory to actively select features for robust speech recognition with interruptions, filtering, and noise," in *Proc. EUROSPEECH*, Rhodes, Greece, Sept. 1997, pp. 37–40.
[33] J. Song and A. Samouelian, "A robust speaker-independent isolated word HMM recognizer for operation over the telephone network," *Speech Commun.*, vol. 13, no. 3/4, pp. 287–295, Dec. 1993.
[34] *Software Tools for Speech and Audio Coding Standardization*, 1996.
[35] K. M. Ponting and S. M. Peeling, "The use of variable frame rate analysis in speech recognition," *Comput. Speech Lang.*, vol. 5, no. 2, pp. 169–179, Apr. 1991.

Hong Kook Kim (M'98–SM'01) received the B.S. degree in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1988, and the the M.S. and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST) in 1990 and 1994, respectively.

He was with Samsung Electronics Company, Ltd., from 1991 to 1993, and Samsung Advanced Institute of Technology (SAIT) from 1993 to 1994, developing a voice dialing system for digital keyphone and implementing the algorithms of speech coding and speech synthesis. From the 1994 to 1998, he was Member of Technical Staff with SAIT, where he developed speech coders operating at 4 kbit/s. In 1998, he was a Senior Researcher and Research Scientist with MMC Technology, Inc., and KAIST, respectively, and he has developed a voice dialing system operating in a mobile environment. From December 1998 to May 2000, he was with AT&T Labs—Research, Florham Park, NJ, as a Consultant in wireless speech recognition; he became Senior Technical Staff Member in June 2000. His current research interests include speech analysis for wireless speech recognition and coding, speech recognition in noisy environments, and speech coding for the Internet and its interoperability in communications networks.

Dr. Kim is a member of KSEA.

Richard V. Cox (M'70–SM'87–F'91) received the Ph.D. degree in electrical engineering from Princeton University, Princeton, NJ.

In 1979, he joined the Acoustics Research Department of Bell Laboratories. He conducted research in the areas of speech coding, digital signal processing, analog voice privacy, audio coding, and real-time implementations. He is well-known for his work in speech coding standards. He collaborated on the low-delay CELP algorithm that became ITU-T Recommendation G.728 in 1992. He managed the ITU effort that resulted in the creation of ITU-T Recommendation G.723.1 in 1995. In 1987, he was promoted to Supervisor of the Digital Principles Research Group. In 1992, he was appointed Department Head of the Speech Coding Research Department of AT&T Bell Labs. In 1996, he joined AT&T Labs as Division Manager of the Speech Processing Software and Technology Research Department. In August 2000, he was appointed Speech and Image Processing Services Research Vice President. In this capacity, he has responsibility for all of AT&T's research in speech, audio, image, video, and multimedia processing research. He is also Vice Chairman of the Board of Directors of Recording for the Blind and Dyslexic (RFB&D), the only U.S. provider of textbooks and reference books for people with print disabilities. At RFB&D, he is presently helping to lead the effort to develop digital books combining audio, text, images, and graphics for their consumers. These "multimedia books" will be available in 2001 for RFB&D K–14 students throughout the United States.

Dr. Cox is President-Elect of the IEEE Signal Processing Society. In 1999, he was awarded the AT&T Science and Technology Medal.