Distributed Speech Processing in MiPad's Multimodal User Interface

Li Deng, *Senior Member, IEEE,* Kuansan Wang, Alex Acero, *Senior Member, IEEE,*
Hsiao-Wuen Hon, *Senior Member, IEEE,* Jasha Droppo, *Member, IEEE,* Constantinos Boulis,
Ye-Yi Wang, *Member, IEEE,* Derek Jacoby, Milind Mahajan, Ciprian Chelba, and Xuedong D. Huang, *Fellow, IEEE*

***Abstract*—This paper describes the main components of MiPad (Multimodal Interactive PAD) and especially its distributed speech processing aspects. MiPad is a wireless mobile PDA prototype that enables users to accomplish many common tasks using a multimodal spoken language interface and wireless-data technologies. It fully integrates continuous speech recognition and spoken language understanding, and provides a novel solution for data entry in PDAs or smart phones, often done by pecking with tiny styluses or typing on minuscule keyboards. Our user study indicates that the throughput of MiPad is significantly superior to that of the existing pen-based PDA interface.**

Acoustic modeling and noise robustness in distributed speech recognition are key components in MiPad's design and implementation. In a typical scenario, the user speaks to the device at a distance so that he or she can see the screen. The built-in microphone thus picks up a lot of background noise, which requires MiPad be noise robust. For complex tasks, such as dictating e-mails, resource limitations demand the use of a client–server (peer-to-peer) architecture, where the PDA performs primitive feature extraction, feature quantization, and error protection, while the transmitted features to the server are subject to further speech feature enhancement, speech decoding and understanding before a dialog is carried out and actions rendered. Noise robustness can be achieved at the client, at the server or both. Various speech processing aspects of this type of distributed computation as related to MiPad's potential deployment are presented in this paper. Recent user interface study results are also described. Finally, we point out future research directions as related to several key MiPad functionalities.

***Index Terms*—Client–server computing, distributed speech recognition, error protection, mobile computing, noise robustness, speech-enabled applications, speech feature compression, speech processing systems.**

I. Introduction

THE GRAPHICAL user interface (GUI) has significantly improved computer human interface by using intuitive real-world metaphors. However, it is still far from achieving the ultimate goal of allowing users to interact with computers without much training. In addition, GUI relies heavily on a graphical display, keyboard and pointing devices that are not always available. Mobile computers have constraints on physical size and battery power, or present limitations due to hands-busy eyes-busy scenarios which make traditional GUI a challenge. Spoken language enabled multimodal interfaces are widely believed to be capable of dramatically enhancing the usability of computers because GUI and speech have complementary strengths. While spoken language has the potential to provide a natural interaction model, the difficulty in resolving the ambiguity of spoken language and the high computational requirements of speech technology have so far prevented it from becoming mainstream in a computer's user interface. MiPad, Multimodal Interactive PAD, is a prototype of a wireless Personal Digital Assistant (PDA) that enables users to accomplish many common tasks using a multimodal spoken language interface (speech + pen + display). A key research goal for MiPad is to seek out appropriate venues for applying spoken language technologies to address the user interface challenges mentioned above. One of MiPad's hardware design concepts is shown in Fig. 1.

MiPad intends to alleviate a prevailing problem of pecking with tiny styluses or typing on minuscule keyboards in today's PDAs by adding speech capability through a built-in microphone. Resembling more like a PDA and less like a telephone, MiPad intentionally avoids speech-only interactions. MiPad is designed to support a variety of tasks such as E-mail, voice-mail, calendar, contact list, notes, web browsing, mobile phone, and document reading and annotation. This collection of functions unifies the various mobile devices into a single, comprehensive communication and productivity tool. The idea is therefore similar to other speech enabled mobile device efforts reported in [3], [16], [21], [24]. While the entire functionality of MiPad can be accessed by pen alone, we found a better user experience can be achieved by combining pen and speech inputs. The user can dictate to an input field by holding the pen down on it. Other pointing devices, such as a roller on the side of the device, device for navigating among the input fields, can also be employed to enable one handed operation. The speech input method, called *Tap & Talk*, not only indicates where the recognized text should go but also serves as a push to talk button. *Tap & Talk* narrows down the number of possible utterances for the spoken language processing module. For example, selecting the "*To:* field" on an e-mail application display indicates that the user is about to enter a name. This dramatically reduces the complexity of spoken language processing and cuts down the speech recognition and understanding errors to the extent that MiPad can be made practically usable despite the current limitations of robust speech recognition and natural language processing technology.

Manuscript received September 25, 2001; revised August 2, 2002. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Harry Printz.

L. Deng, K. Wang, A. Acero, H.-W. Hon, J. Droppo, Y.-Y. Wang, D. Jacoby, M. Mahajan, C. Chelba, and X. D. Huang are with Microsoft Research, Redmond, WA 98052 USA (e-mail: deng@microsoft.com).

C. Boulis is with the University of Washington, Seattle, WA 98195 USA.

Digital Object Identifier 10.1109/TSA.2002.804538

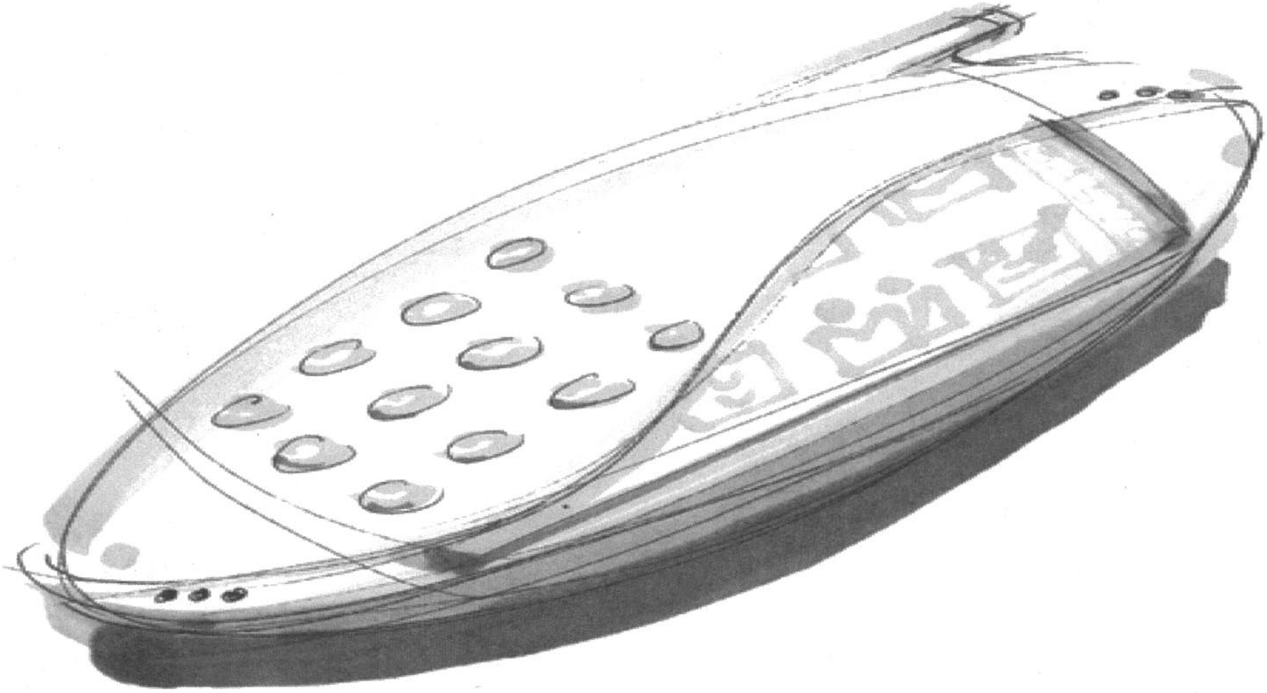

Fig. 1. One of MiPad's industrial design templates.

One key feature of MiPad is a general purpose "Command" field to which a user can issue naturally spoken commands such as "Schedule a meeting with Bill tomorrow at two o'clock." From the user's perspective, MiPad not only recognizes but *understands* the command by MiPad executing the necessary actions conveyed in the spoken commands. In response to the above command, MiPad will display a "meeting arrangement" screen with related fields (such as date, time, attendees, etc.) filled appropriately based on the user's utterance. MiPad fully implements Personal Information Management (PIM) functions including email, calendar, notes, task, and contact list with a hardware prototype based on Compaq's iPaq PDA (3800 series). All MiPad applications are configured in a client–server architecture as shown in Fig. 2. The client on the left side of Fig. 2 is MiPad powered by Microsoft Windows CE operating system that supports 1) sound capture, 2) front-end acoustic processing including noise reduction, channel normalization, feature compression, and error protection, 3) GUI processing, and 4) a fault-tolerant communication layer that allows the system to recover gracefully from network connection failures. Specifically, to reduce bandwidth requirements, the client compresses the wideband speech parameters down to a maximal 4.8 Kbps bandwidth. Between 1.6 and 4.8 Kbps, we observed virtually no increase in the recognition error on some tasks tested. A wireless local area network (WLAN), which is currently used to simulate a third generation (3G) wireless network, connects MiPad to a host machine (server) where the continuous speech recognition (CSR) and spoken language understanding (SLU) take place. The client takes approximately 450 KB of program space and an additional 200 KB of runtime heap, and merely consumes approximately 35% of CPU load with iPAQ's 206 MHz StrongARM processor. At the server side, as shown on the right side of Fig. 2, MiPad applications communicate with the ASR and SLU engines for coordinated context-sensitive *Tap & Talk* interaction. Noise robustness processing also takes place at the server since it allows for easy updating.

We now describe the rationale behind MiPad's architecture. Although customized system software and hardware have been reported [3], [16] to bring extra benefits and flexibility in tailoring applications to mobile environments, the MiPad project utilizes only off-the-shelf hardware and software. Given the rapid improvements in the hardware and system software capabilities, we believe such an approach is a reasonable one. Second, although speaker independent speech recognition has made significant strides during the past two decades, we have deliberately positioned MiPad as a *personal* device where the user profile can be utilized to enrich applications and complement technological shortcomings. For speech, this means we may use speaker dependent recognition, thereby avoiding the challenges faced by other approaches [21], [24]. In addition to enabling higher recognition accuracy, user specific information can also be stored locally and speaker specific processing can be carried out on the client device itself. This architecture allows us to create user customized applications using generic servers, thereby improving overall scalability.

The rest of the paper will describe details of MiPad with emphasis on the speech processing and the UI design considerations. Various portions of this paper have been presented at several conferences (ICSLP-2000 [1], [18], [28] ICASSP-2001 [5], [19], [25], Eurospeech-2001 [12] and ASRU-2001 Workshop [6]). The purpose of this paper is to combine these earlier presentations on the largely isolated MiPad components into a single coherent paper so as to highlight the important roles of distributed speech processing in the MiPad design, and report some more recent research results. The organization of this paper is as follows. In Sections II and III, we describe our re-

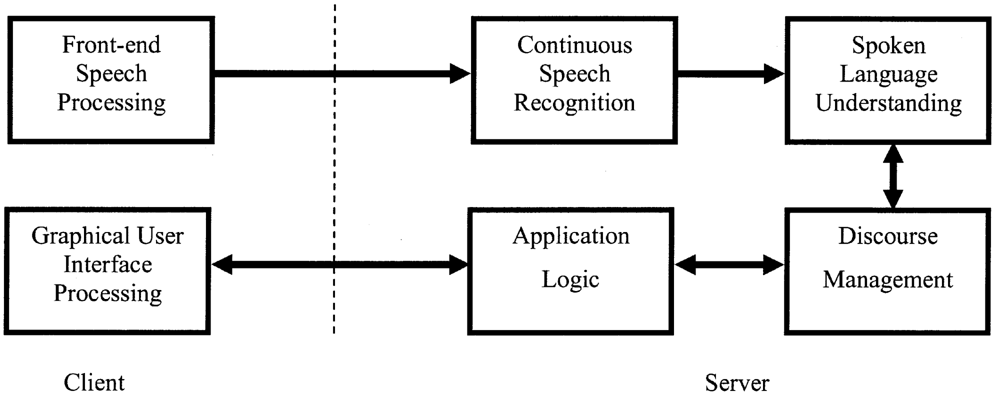
Front-end Speech Processing
Continuous Speech Recognition
Spoken Language Understanding
Graphical User Interface Processing
Application Logic
Discourse Management
Client
Server

Fig. 2. MiPad's client–server (peer-to-peer) architecture. The client is based on a Windows CE iPAQ, and the server is based on a Windows server. The client–server communication is currently implemented on a wireless LAN.

cent work on front-end speech processing, including noise robustness and source/channel coding that underlie MiPad's distributed speech recognition capabilities. The acoustic and language models used for the decoding phase of MiPad continuous speech recognition, together with the spoken language understanding component, are presented in Section IV. Finally, MiPad's user interface and user study results are described in Section V, and a summary is provided in Section VI.

II. Robustness to Acoustic Environments

Immunity to noise and channel distortion is one of the most important design considerations for MiPad. For this device to be acceptable to the general public, it is desirable to remove the need for a close-talking microphone. However, with the convenience of using the built-in microphone, noise robustness becomes a key challenge to maintaining desirable speech recognition and understanding performance. Our recent work on acoustic modeling for MiPad has focused on overcoming this noise-robustness challenge. In this section we will present most recent results in the framework of distributed speech recognition (DSR) that the MiPad design has adopted.

A. Distributed Speech Recognition Considerations for Algorithm Design

There has been a great deal of interest recently in standardizing DSR applications for a plain phone, PDA, or a smart phone where speech recognition is carried out at a remote server. To overcome bandwidth and infrastructure cost limitations, one possibility is to use a standard codec on the device to transmit the speech to the server where it is subsequently decompressed and recognized. However, since speech recognizers such as the one in MiPad only need some features of the speech signal (e.g., Mel-cepstrum), bandwidth can be further saved by transmitting only those features. ETSI has been accepting proposals for Aurora [17], an effort to standardize a DSR front-end that addresses the issues surrounding robustness to noise and channel distortions at a low bit rate. Our recent work on noise robustness for MiPad has been concentrated on the Aurora tasks.

In DSR applications, it is easier to update software on the server because one cannot assume that the client is always running the latest version of the algorithm. With this consideration in mind, while designing noise-robust algorithms for MiPad, we strive to make the algorithms front-end agnostic. That is, the algorithms should make no assumptions on the structure and processing of the front end and merely try to undo whatever acoustic corruption has been shown during training. This consideration also favors approaches in the feature rather than the model domain.

Here, we describe one particular algorithm that has so far given the best performance on the Aurora2 task and other Microsoft internal tasks. We called the algorithm SPLICE, short for Stereo-based Piecewise Linear Compensation for Environments. In a DSR system, SPLICE may be applied either within the front end on the client device, or on the server, or on both with collaboration. Certainly a server side implementation has some advantages as computational complexity becomes less of an issue and continuing improvements can be made to benefit even devices already deployed in the field. Another useful property of SPLICE in the server implementation is that new noise conditions can be added as they are identified by a server. This can make SPLICE quickly adaptable to any new acoustic environment with minimum additional resources.

B. Basic Version of SPLICE

SPLICE is a frame-based, bias removal algorithm for cepstrum enhancement under additive noise, channel distortion or a combination of the two. In [4], we reported the approximate MAP formulation of the algorithm, and more recently in [5], [11], [12] we described the MMSE formulation of the algorithm with a much wider range of naturally occurring noises, including both artificially mixed speech and noise, and naturally recorded noisy speech.

SPLICE assumes no explicit noise model, and the noise characteristics are embedded in the piecewise linear mapping between the "stereo" clean and distorted speech cepstral vectors. The piecewise linearity is intended to approximate the true nonlinear relationship between the two. The nonlinearity between the clean and distorted (including additive noise) cepstral vectors arises due to the use of the logarithm in computing the cepstra. Stereo data refers to simultaneously recorded waveforms both on clean and noisy speech. SPLICE is potentially able to handle a wide range of distortions, including nonstationary distortion, joint additive and convolutional distortion, and nonlinear distortion (in time-domain) because the stereo data provides accurate estimates of the bias or correction vectors without the need for an explicit noise model. One key requirement for the success of the basic version of SPLICE described here is that the distortion conditions under which the correction vectors are learned from the stereo data must be similar to those corrupting the test data. Enhanced versions of the algorithm described later in this section will relax this requirement by employing a noise estimation and normalization procedure.

We assume a general nonlinear distortion of a clean cepstral vector, $\mathbf{x}$, into a noisy one, $\mathbf{y}$. This distortion is approximated in SPLICE by a set of linear distortions. The probabilistic formulation of the basic version of SPLICE is provided below.

1) Basic Assumptions: The first assumption is that the noisy speech cepstral vector follows a mixture distribution of Gaussians

$$p(\mathbf{y}) = \sum_s p(\mathbf{y}|s)p(s), \quad \text{with} \quad p(\mathbf{y}|s) = N(\mathbf{y};\, \boldsymbol{\mu}_s,\, \boldsymbol{\Sigma}_s) \quad (1)$$

where s denotes the discrete random variable taking the values $1, 2, \ldots, N$, one for each region over which the piecewise linear approximation between the clean cepstral vector $\mathbf{x}$ and distorted cepstral vector is made. This distribution, one for each separate distortion condition (not indexed for clarity), can be thought as a "codebook" with a total of N codewords (Gaussian means) and their variances.

The second assumption made by SPLICE is that the conditional probability density function (PDF) for the clean vector $\mathbf{x}$ given the noisy speech vector, $\mathbf{y}$, and the region index, s, is a Gaussian with the mean vector being a linear function of the noisy speech vector $\mathbf{y}$. In this paper, we take a simplified form of this (piecewise) function by making the rotation matrix to be identity one, leaving only the bias or correction vector. Thus, the conditional PDF has the form

$$p(\mathbf{x}|\mathbf{y},\, s) = N(\mathbf{x}; \mathbf{y} + \mathbf{r}_s,\, \boldsymbol{\Gamma}_s) \quad (2)$$

where the correction vector is $\mathbf{r}_s$ and the covariance matrix of the conditional PDF is $\boldsymbol{\Gamma}_s$.

2) SPLICE Training: Since the noisy speech PDF $p(\mathbf{y})$ obeys a mixture-of-Gaussian distribution, the standard EM algorithm is used to train $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$. Initial values of the parameters can be determined by a VQ clustering algorithm.

The parameters $\mathbf{r}_s$ and $\boldsymbol{\Gamma}_s$ of the conditional PDF $p(\mathbf{x}|\mathbf{y},\, s)$ can be trained using the maximum likelihood criterion. Since the variance of the distribution is not used in cepstral enhancement, we only give the ML estimate of the correction vector below:

$$\mathbf{r}_s = \frac{\sum_n p(s|\mathbf{y}_n)(\mathbf{x}_n - \mathbf{y}_n)}{\sum_n p(s|\mathbf{y}_n)} \quad (3)$$

where

$$p(s|\mathbf{y}_n) = \frac{p(\mathbf{y}_n|s)p(s)}{\sum_r p(\mathbf{y}_n|r)p(r)} \quad (4)$$

and n denotes the time-frame index of the feature vector.

This training procedure requires a set of stereo (two channel) data. One channel contains the clean utterance, and the other channel contains the same utterance with distortion,. The two-channel data can be collected, for example, by simultaneously recording utterances with one close-talking and one far-field microphone. Alternatively, it has been shown in our research [10] that a large amount of synthetic stereo data can be effectively produced to approximate the realistic stereo data with virtually no loss of speech recognition accuracy in MiPad tasks.

3) SPLICE for Cepstral Enhancement: One significant advantage of the above two basic assumptions made in SPLICE is the inherent simplicity in deriving and implementing the rigorous MMSE estimate of clean speech cepstral vectors from their distorted counterparts. Unlike the FCDCN algorithm [1], no approximations are made in deriving the optimal enhancement rule. The derivation is outlined below.

The MMSE is the following conditional expectation of clean speech vector given the observed noisy speech:

$$E_x[\mathbf{x}|\mathbf{y}] = \sum_s p(s|\mathbf{y}) E_{\mathbf{x}}[\mathbf{x}|\mathbf{y},\, s]. \quad (5)$$

Due to the second assumption of SPLICE, the above codeword-dependent conditional expectation of $\mathbf{x}$ (given $\mathbf{y}$ and s) is simply the bias-added noisy speech vector

$$E_x[\mathbf{x}|\mathbf{y},\, s] = \mathbf{y} + \mathbf{r}_s \quad (6)$$

where bias $\mathbf{r}_s$ has been estimated from the stereo training data according to (3). This gives the simple form of the MMSE estimate as the noisy speech vector corrected by a linear weighted sum of all codeword-dependent bias vectors already trained

$$\hat{\mathbf{x}} = E_x[\mathbf{x}|\mathbf{y}] = \mathbf{y} + \sum_s p(s|\mathbf{y})\mathbf{r}_s. \quad (7)$$

While this is already efficient to compute, more efficiency can be achieved by approximating the weights according to

$$\hat{p}(s|\mathbf{y}) \cong \begin{cases} 1, & s = \arg\max_s p(s|\mathbf{y}) \\ 0, & \text{otherwise.} \end{cases} \quad (8)$$

This approximation turns the MMSE estimate to the approximate MAP estimate that consists of two sequential steps of operation. First, finding optimal codewords using the VQ codebook based on the parameters $(\mathbf{r}_s,\, \boldsymbol{\Gamma}_s)$, and then adding the codeword-dependent vector $\mathbf{r}_s$ to the noisy speech vector. We have found empirically in many of our initial experiments that the above VQ approximation does not appreciably affect recognition accuracy while resulting in computational efficiency.

C. Enhancing SPLICE by Temporal Smoothing

In this enhanced version of SPLICE, we not only minimize the static deviation from the clean to noisy cepstral vectors (as in the basic version of SPLICE), but also seek to minimize the dynamic deviation.

The basic SPLICE optimally processes each frame of noisy speech independently. An obvious extension is to jointly process a segment of frames. In this way, although the deviation from the clean to noisy speech cepstra for an individual frame could be undesirably greater than that achieved by the basic, static

SPLICE, the overall deviation that takes into account the whole sequence of frames and the mismatch of slopes will be reduced compared with the basic SPLICE.

We have implemented the above idea of "dynamic SPLICE" through temporally smoothing the bias vectors obtained from the basic, static SPLICE.

We have achieved significant performance gains using an efficient heuristic implementation. In our specific implementation, the filter has a low-pass characteristic, with a system transfer function of

$$H(z) = \frac{-0.5}{(z^{-1} - 0.5)(z - 2)}. \tag{9}$$

This transfer function is the result of defining an objective function as the posterior probability of the entire sequence of the (hidden) true correction vectors given the entire sequence of the observed speech vectors. The posterior probability is

$$p(\mathbf{r}_1, \ldots, \mathbf{r}_T | \mathbf{y}_1, \ldots, \mathbf{y}_T).$$

After we apply a first-order Markov assumption to the $\mathbf{r}_n$, this conditional distribution becomes

$$p(\mathbf{r}_1|\mathbf{y}_1) \prod_{n=2}^{T} p(\mathbf{r}_n|\mathbf{r}_{n-1}, \mathbf{y}_n). \tag{10}$$

Each term in the product is given by

$$p(\mathbf{r}_n|\mathbf{r}_{n-1}, \mathbf{y}_n) = N(\mathbf{r}_n; \mathbf{y}_n - \hat{\mathbf{x}}_n, \mathbf{\Sigma}_{\hat{\mathbf{x}}_n}) N(\mathbf{r}_n; \mathbf{r}_{n-1}, \mathbf{\Sigma}_{\Delta})$$

where $\mathbf{\Sigma}_{\Delta}$ is the covariance matrix for the time differential of the correction vector, $\hat{\mathbf{x}}_n$ is the unsmoothed SPLICE output at frame n, given by (7), and $\mathbf{\Sigma}_{\hat{\mathbf{x}}_n}$ is the covariance matrix of the SPLICE output at frame n given by

$$\mathbf{\Sigma}_{\hat{\mathbf{x}}_n} = E[(\mathbf{x}_n - \hat{\mathbf{x}}_n)(\mathbf{x}_n - \hat{\mathbf{x}}_n)^* | \mathbf{y}_n].$$

Optimization of the objective function in (10) gives the MAP estimate for the smoothed correction vector sequence, which is in the form of a second-order difference equation with the input of unsmoothed correction vectors computed from the SPLICE algorithm described above. This second-order difference equation can be equivalently put in the form of (9) in the z-domain, where the constants are functions of the variances related to both the static and dynamic quantities of the correction vectors. These variances were assumed to be time invariant, leading to the two constant parameters in (9). These two parameters have been empirically adjusted.

D. Enhancing SPLICE by Noise Estimation and Noise Normalization

In this enhancement of SPLICE, different noise conditions between the SPLICE training set and test set are normalized. The procedure for noise normalization and for denoising is as follows. Instead of building codebooks for noisy speech $\mathbf{y}$ from the training set, they are built from $\mathbf{y} - \mathbf{n}$ where $\mathbf{n}$ is an estimated noise from $\mathbf{y}$. Then the correction vectors are estimated from the training set using the noise-normalized stereo data $(\mathbf{y} - \mathbf{n})$ and $(\mathbf{x} - \mathbf{n})$. The correction vectors trained in this new SPLICE will be different from those in the basic version of SPLICE. This is because the codebook selection will be different since $p(s|\mathbf{y})$ is changed to $p(s|\mathbf{y} - \mathbf{n})$. For denoising in the test data, the noise-normalized noisy cepstra $\mathbf{y} - \mathbf{n}$ are used to obtain the noise-normalized MMSE estimate, and then the noise normalization is undone by adding the estimated noise $\mathbf{n}$ back to the MMSE estimate.

Our research showed that the effectiveness of the above noise-normalized SPLICE is highly dependent on the accuracy of the noise estimate $\mathbf{n}$. We have investigated several ways of automatically estimating the nonstationary noise in the Aurora2 database. We describe below one algorithm that has given by far the highest accuracy in noise estimation and at the same time by far the best noise-robust speech recognition results.

E. Nonstationary Noise Estimation by Iterative Stochastic Approximation

In [6], a novel algorithm is proposed, implemented, and evaluated for recursive estimation of parameters in a nonlinear model involving incomplete data. The algorithm is applied specifically to time-varying deterministic parameters of additive noise in a mildly nonlinear model that accounts for the generation of the cepstral data of noisy speech from the cepstral data of the noise and clean speech. For computer recognition of the speech that is corrupted by highly nonstationary noise, different observation data segments correspond to very different noise parameter values. It is thus strongly desirable to develop recursive estimation algorithms, since they can be designed to adaptively track the changing noise parameters. One such design based on the novel technique of iterative stochastic approximation within the recursive-EM framework is developed and evaluated. It jointly adapts time-varying noise parameters and the auxiliary parameters introduced to piecewise linearly approximate the nonlinear model of the acoustic environment. The accuracy of the approximation is shown to have improved progressively with more iteration.

The essence of the algorithm is the use of iterations to achieve close approximations to a nonlinear model of the acoustic environment while at the same time employing the "forgetting" mechanism to effectively track nonstationary noise. There is no latency required for the execution of the algorithm since only the present and the past noisy speech observations are needed to compute the current frame's noise estimate. Using a number of empirically verified assumptions associated with the implementation simplification, the efficiency of this algorithm has been improved close to real time for noise tracking. The mathematical theory, algorithm, and implementation detail of this iterative stochastic approximation technique can be found in [6], [7].

Figs. 3–5 show the results of noise-normalized SPLICE denoising using the iterative stochastic algorithm for tracking nonstationary noise $\mathbf{n}$ in an utterance of the Aurora2 data, where the SNR is 10 dB, 5 dB, and 0 dB, respectively. From top to bottom we can see noisy speech, clean speech, and denoised speech, all in the same spectrogram format. Most of the noise has been effectively removed, except for some strong noise burst located around frames 150–158 in Fig. 5 where the instantaneous SNR is significantly lower than zero.

The nonstationary noisy estimation algorithm discussed here and its use in the noise-normalized SPLICE are critical factors for the noise-robust speech recognition results presented in the next section. We have recently extended the algorithm to represent the noise as time-varying random vectors in order to exploit the variance parameter and new prior information. The

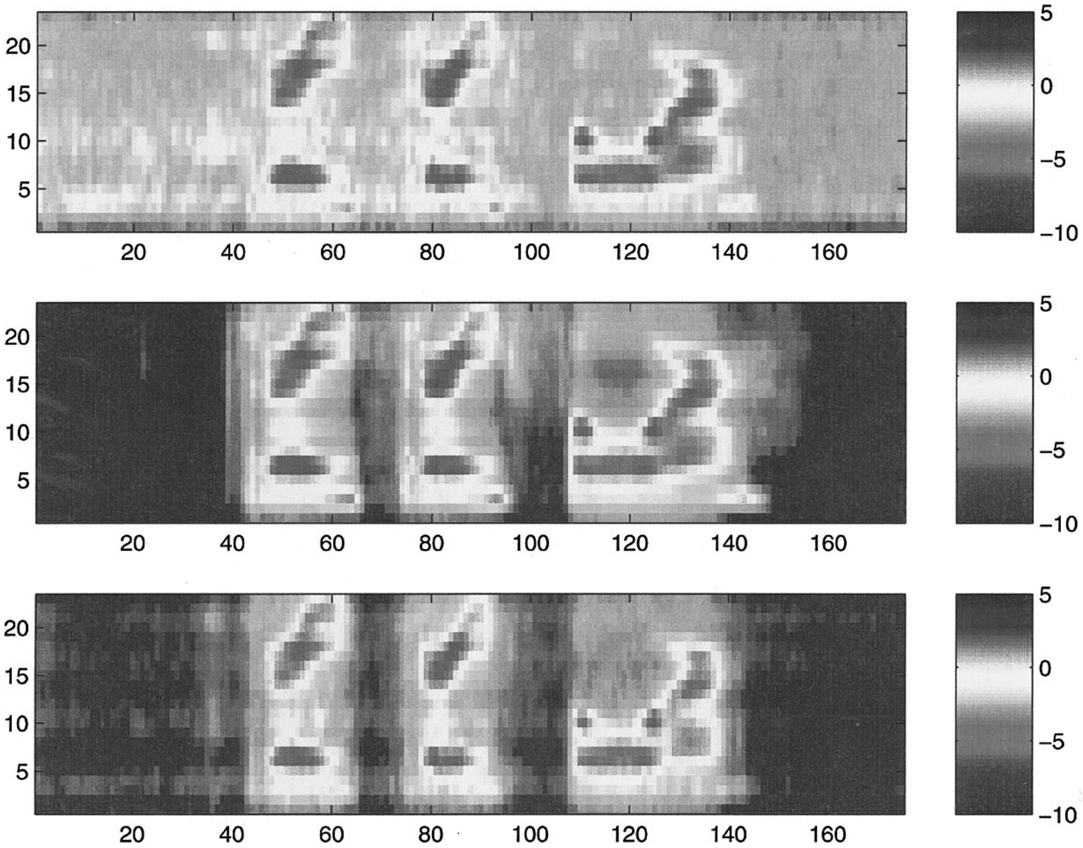

Fig. 3. Noise-normalized SPLICE denoising using the iterative stochastic algorithm for tracking nonstationary noise in an utterance of the Aurora2 data with an average SNR $=$ 10 dB. From top to bottom panels are noisy speech, clean speech, and denoised speech, all in the same spectrogram format.

algorithm has also been successfully extended from the maximum likelihood version to the new MAP version to take advantage of the noise prior information and to include a more accurate environment model that captures more detailed properties of acoustic distortion [8].

F. Aurora2 Evaluation Results

Noise-robust connected digit recognition results obtained using the best version of SPLICE are shown in Fig. 6 for the full Aurora2 evaluation test data. Details of the Aurora2 task have been described in [17]. Aurora2 is based on the TIDigits database that is corrupted digitally by adding different types of realistic, nonstationary noises at a wide range of SNRs (all Sets A, B, and C) and optionally passing them through a linear filter (Set C only). Sets-A and -B each consists of 1101 digit sequences for each of four noise conditions and for each of the 0 dB, 5 dB, 10 dB, 15 dB, and 20 dB SNRs. The same is for Set-C except there are only two noise conditions. All the results in Fig. 6 are obtained with the use of cepstral mean normalization (CMN) for all data after applying noise-normalized, dynamic SPLICE to cepstral enhancement. The use of CMN has substantially improved the recognition rate for Set-C. For simplicity, we have assumed no channel distortion in the implementation of the iterative stochastic approximation algorithm for noise estimation. This assumption would not be appropriate for Set-C which contains unknown but fixed channel distortion. This deficiency has been, at least partially, offset by the use of CMN. All the recognition experiments reported here were obtained using the standard Aurora recognition system [17] instead of our internal recognition system.

The word error rate reduction achieved as shown in Fig. 6 is 27.9% for the multicondition training mode, and 67.4% for the clean-only training mode, respectively, compared with the results using the standard Mel cepstra with no speech enhancement. In the multicondition training mode, the denoising algorithm is applied to the training data set and the resulting denoised Mel-cepstral features are used to train the HMMs. In the clean-only training mode, the HMMs are trained using clean speech Mel-cepstra and the denoising algorithm is applied only to the test set. The results in Fig. 6 represent the best performance in the September-2001 Aurora2 evaluation in the category of the clean speech training mode [20]. The experimental results also demonstrated the crucial importance of using the newly introduced iterations in improving the earlier stochastic approximation technique, and showed a varying degree of sensitivity, depending on the degree of noise nonstationarity, of the noise estimation algorithm's performance to the forgetting factor embedded in the algorithm [7]. More recently, the success of the noise-normalized SPLICE algorithm has been extended from the Aurora2 task to the Aurora3 task [11].

There has been a wide range of research groups around the world working on the same problem of noise-robust speech recognition for mobile and other devices as we are interested in; see [2], [9], [14], [15], [20], [22], [23] for selected approaches taken by some of these research groups. The general

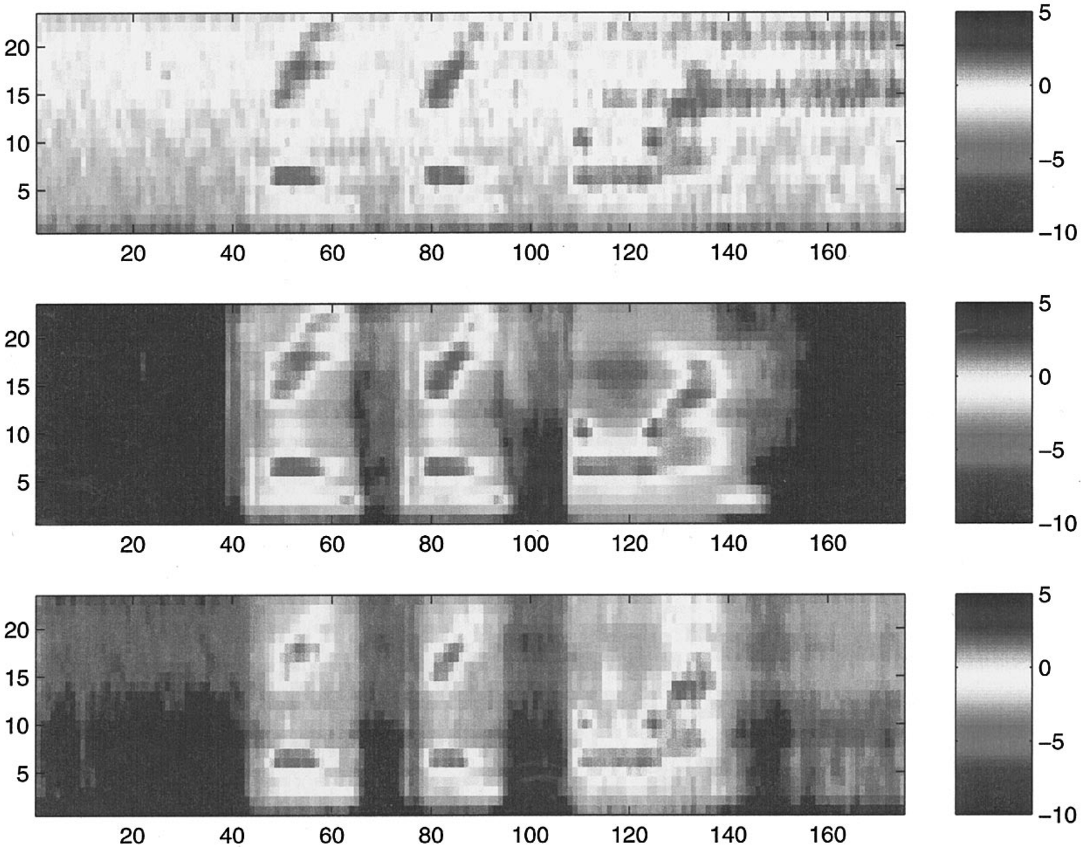

Fig. 4. Noise-normalized SPLICE denoising using the iterative stochastic algorithm for tracking nonstationary noise in an utterance of the Aurora2 data with an average SNR = 5 dB. From top to bottom panels are noisy speech, clean speech, and denoised speech, all in the same spectrogram format.

approaches taken can be classified into either the model-domain or the feature-domain one, with respective strengths and weaknesses. The approach reported in this paper is unique in that it takes full advantage of the rich information embedded in the stereo training data which most directly characterize the relationship between the clean and noise speech feature vectors. The approach is also unique in that a powerful noise tracking algorithm is exploited to effectively compensate for possible mismatch between the operating condition and the condition under which the SPLICE parameters are trained. The feature-domain approach we have taken is based on the special DSR considerations for MiPad architecture.

III. Feature Compression and Error Protection

In addition to noise robustness, we recently also started work on feature compression (source coding) and error protection (channel coding) required by MiPad's client–server architecture. This work is intended to address the three key requirements for successful deployment of distributed speech recognition associated with the client–server approach: 1) compression of cepstral features (via quantization) must not degrade speech recognition performance; 2) the algorithm for source and channel coding must be robust to packet losses, bursty or otherwise; and 3) the total time delay due to the coding, which results from a combined quantization delay, error-correction coding delay, and transmission delay, must be kept within an acceptable level. In this section, we outline the basic approach and preliminary results of this work.

A. Feature Compression

A new source coding algorithm has been developed that consists of two sequential stages. After the standard Mel-cepstra are extracted, each speech frame is first classified to a phonetic category (e.g., phoneme) and then is vector quantized (VQ) using the split-VQ approach. The motivation behind this new source coder is that the speech signal can be composed of piecewise-stationary segments, and therefore can be most efficiently coded using one of many small codebooks that is tuned into a particular segment. Also, the purpose of the source coding considered here is to reduce the effect of coding on the speech recognizer's word error rate on the server-side of MiPad, which is very different from the usual goal of source coding aiming at maintaining perceptual quality of speech. Therefore, the use of phone-dependent codebooks is deemed most appropriate since phone distinction can be enhanced by using separate codebooks for distinct phones. Phone distinction often leads to word distinction, which is the goal of speech recognition and also the ultimate goal of the feature compression in MiPad.

One specific issue to be addressed in the coder design is bit allocation, or the number of bits that must be assigned to the subvector codebooks. In our coder, C0, C1–6, and C7–12 are $M = 3$ separate subvectors that are quantized independently. Starting from 0 bits for each subvector codebook of each phone

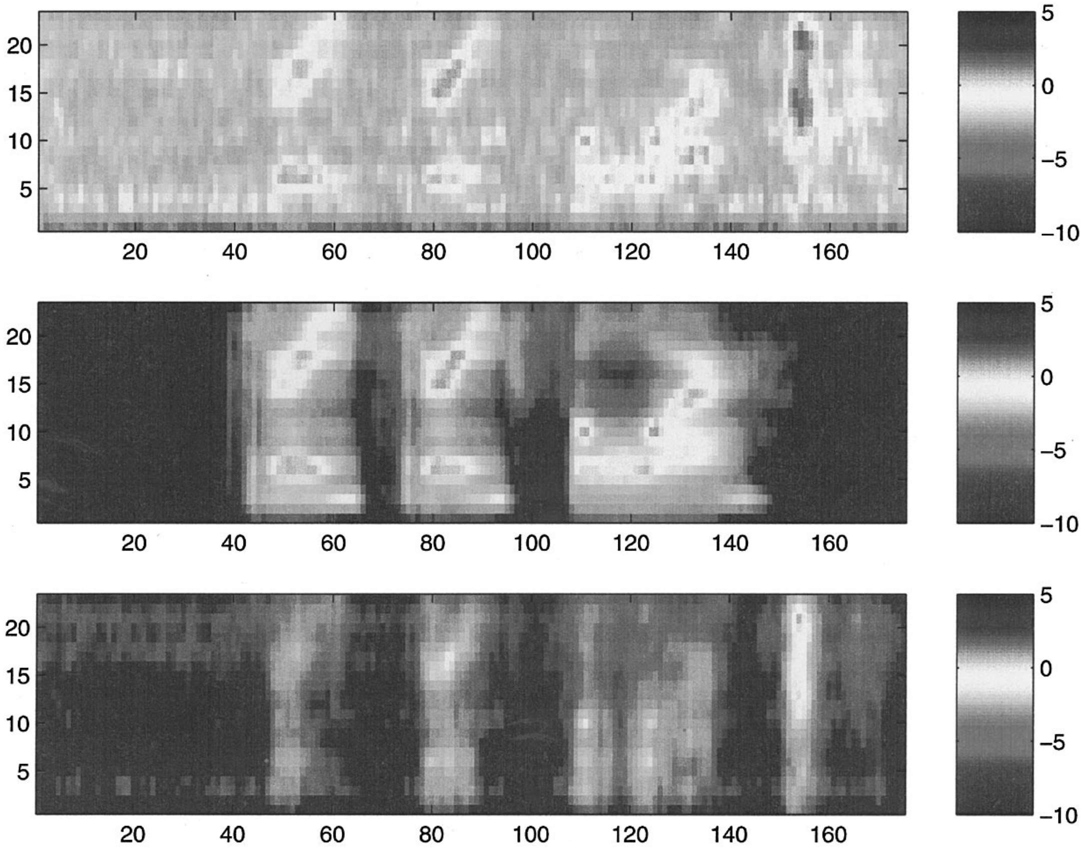

Fig. 5. Noise-normalized SPLICE denoising using the iterative stochastic algorithm for tracking nonstationary noise in an utterance of the Aurora2 data with an average SNR $= 0$ dB. From top to bottom panels are noisy speech, clean speech, and denoised speech, all in the same spectrogram format.

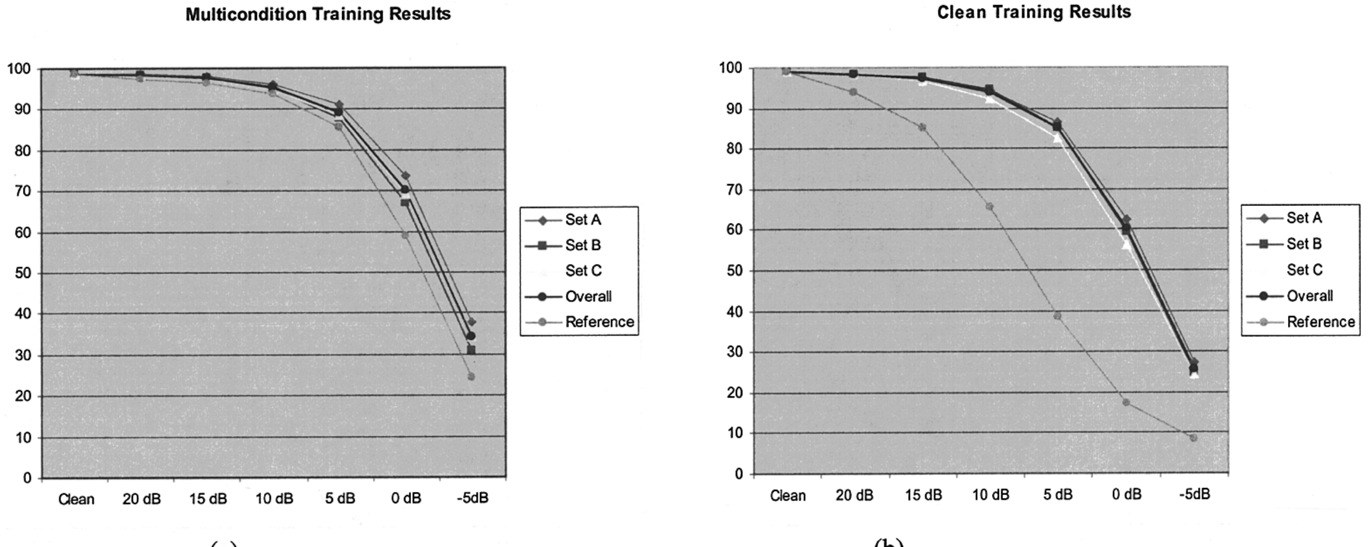

Fig. 6. Full set of noise-robust speech recognition results in the September-2001 Aurora2 evaluation, using the dynamic and noise-normalized SPLICE with the noise estimation obtained from iterative stochastic approximation; Sets A, B, and C are separate test sets with different noise and channel distortion conditions. In (a) are the recognition rates using multicondition training mode where the denoising algorithm is applied to the training data set and the resulting denoised Mel-cepstral features are used to train the HMMs. In (b) are the recognition rates using the "clean" training model where the HMMs are trained using clean speech Mel-cepstra and the denoising algorithm is applied only to the test set. Reference curves in both (a) and (b) refer to the recognition rates obtained with no denoising processing.

we can evaluate every possible combination of bits to subvectors and select the best according to a certain criterion. To better match the training procedure to the speech recognition task we use the criterion of minimal word error rate (WER). That is, bit assignment is the result of the following constrained optimization:

$$B = \arg\min_{B}\{WER(B)\} \tag{11}$$

under the constraint of $\sum_i b_i = N$, where b_i is the number of bits assigned to the ith subvector, N is the total number of bits to be assigned, and $WER(B)$ is the WER obtained using B as the assignment of the bits to subvectors. Full search requires us to run a separate WER experiment for each one of the possible combinations, computationally prohibitive, so we use a greedy bit allocation technique. At each stage we add a bit at each one of the subvectors and we keep the combination with the minimal WER. We repeat the procedure for the next stage by starting at the best combination of the previous step. By having M subvectors and N total bits to assign the total number of combinations is reduced from M^N to $M \times N$.

The experiments carried out to evaluate the above phone-dependent coder use the baseline system with a version of Microsoft's continuous-density HMM Whisper system. The system uses 6000 tied HMM states (senones), 20 Gaussians per state, Mel-cepstrum, delta cepstrum, and delta–delta cepstrum. The recognition task is 5000-word vocabulary, continuous speech recognition from Wall Street Journal data sources. A fixed, bigram language model is used in all the experiments. The training set consists of a total of 16 000 female sentences, and the test set of 167 female sentences (2708 words). The word accuracy with no coding for this test set was 95.7%. With use of a perfect phone classifier, the coding using the bit allocation of (4, 4, 4) for the three subvectors gives word accuracy of 95.6%. Using a very simple phone classifier with a Mahalanobis distance measure and with the same bit allocation of (4, 4, 4), the recognition accuracy drops only to 95.0%. For this high-performance coder, the bandwidth has been reduced to 1.6 Kbps with the required memory being under 64 Kbytes.

B. Error Protection

A novel channel coder has also been developed to protect MiPad's Mel-cepstral features based on the client–server architecture. The channel coder assigns unequal amounts of redundancy among the different source bits, giving a greater amount of protection to the most important bits where the importance is measured by the contributions of these bits to the word error rate in speech recognition. A quantifiable procedure to assess the importance of each bit is developed, and the channel coder exploits this utility function for the optimal forward error correction (FEC) assignment. The FEC assignment algorithm assumes that packets are lost according to a Poisson process. Simulation experiments are performed where the bursty nature of loss patterns are taken into account. When combined with the new source coder, the new channel coder is shown to provide considerable robustness to packet losses even under extremely adverse conditions.

Some alternatives to FEC coding are also explored, including the use of multiple transmissions, interleaving, and interpolation. We conclude from this preliminary work that the final choice of channel coder should depend on the relative importance among delay, bandwidth, and burstiness of noise.

Our preliminary work on the compression and error protection aspects of distributed speech recognition has provided clear insight into the tradeoffs we need to make between source coding, delay, computational complexity and resilience to packet losses. Most significantly, the new algorithms developed have brought down the Mel-cepstra compression rate to as low as 1.6 Kbps with virtually no degradation in word error rate compared with no compression. These results are currently being incorporated into the next version of MiPad.

IV. Continuous Speech Recognition and Understanding

While the compressed and error-protected Mel-cepstral features are computed in the MiPad client, major computation for continuous speech recognition (decoding) resides in the server. The entire set of the language model, hidden Markov models (HMMs), and lexicon that are used for speech decoding all reside in the server, which processes the Mel-cepstral features transmitted from the client. Denoising operations such as SPLICE that extract noise-robust Mel-cepstra can reside either on the server or the client, though we implemented it on the server for convenience.

MiPad is designed to be a personal device. As a result, speech recognition uses speaker-adaptive acoustic models (HMMs) and a user-adapted lexicon to improve recognition accuracy. The continuous speech recognition engine and its HMMs are a hybrid that combines the best features of Microsoft's Whisper and HTK. Both MLLR and MAP adaptation are used to adapt the speaker-independent acoustic model for each individual speaker. We used 6000 senones, each with 20-component mixture Gaussian densities. The context-sensitive language model is used for relevant semantic objects driven by the user's pen tapping action, as described in Section IV. As speech recognition accuracy remains as a major challenge for MiPad usability, most of our recent work on MiPad's acoustic modeling has focused on noise robustness as described in Section II. The work on language modeling for improving speech recognition accuracy has focused on language model portability, which is described in this section.

The speech recognition engine in MiPad uses the unified language model [25] that takes advantage of both rule-based and data-driven approaches. Consider two training sentences:

"Meeting at three with John Smith." versus
"Meeting at four PM with Derek."

Within a pure n-gram framework, we need to estimate

$$P(John|three\ with) \quad \text{and} \quad P(Derek|PM\ with)$$

individually. This makes it very difficult to capture the obviously needed long-span semantic information in the training data. To overcome this difficulty, the unified model uses a set of Context Free Grammars (CFGs) that captures the semantic structure of the domain. For the example listed here, we may have CFGs for ⟨NAME⟩ and ⟨TIME⟩ respectively, which can be derived from the factoid grammars of smaller sizes. The training sentences now look like:

"Meeting⟨at three:TIME⟩with⟨John Smith:NAME⟩,"
and
"Meeting⟨at four PM:TIME⟩with⟨Derek: NAME⟩."

With parsed training data, we can now estimate the n-gram probabilities as usual. For example, the replacement of

$$P(John|three\ with) \leftarrow P(\langle NAME\rangle|\langle TIME\rangle with)$$

makes such "n-gram" representation more meaningful and more accurate.

TABLE I
CROSS-DOMAIN SPEAKER-INDEPENDENT SPEECH RECOGNITION PERFORMANCE WITH THE UNIFIED LANGUAGE MODEL AND ITS CORRESPONDING DECODER

Systems	Perplexity	Word Error	Relative Decoding Time
Domain-independent Trigram	593	35.6%	1.0
Unified decoder with the unified LM	141	22.5%	0.77
N-best re-scoring with the unified LM		24.2%	

Inside each CFG, however, we can still derive

$$P(\text{“}John\ Smith\text{”}|\langle NAME\rangle) \text{ and } P(\text{“}four\ PM\text{”}|\langle TIME\rangle)$$

from the existing n-gram (n-gram probability inheritance) so that they are appropriately normalized [25]. This unified approach can be regarded as a generalized n-gram in which the vocabulary consists of words and structured classes. The structured class can be simple, such as ⟨DATE⟩, ⟨TIME⟩, and ⟨NAME⟩, if there is no need to capture deep structural information. It can be made complicated also in order to contain deep structured information. The key advantage of the unified language model is that we can author limited CFGs for each new domain and embed them into the domain-independent n-grams. In short, CFGs capture domain-specific structural information that facilitates language model portability, while the use of n-grams makes the speech decoding system robust against catastrophic errors.

Most decoders can only support either CFGs or word n-grams. These two ways of representing sentence probabilities were mutually exclusive. We modified our decoder so that we can embed CFGs in the n-gram search framework to take advantage of the unified language model. An evaluation of the use of the unified language model is shown in Table I. The speech recognition error rate with the use of the unified language model is demonstrated to be significantly lower than that with the use of the domain-independent trigram. That is, incorporating the CFG into the language model drastically improves cross-domain portability. The test data shown in Table I are based on MiPad's PIM conversational speech. The domain-independent trigram language model is based on Microsoft Dictation trigram models used in Microsoft Speech SDK 4.0. In Table I, we also observe that using the unified language model directly in the decoding stage produces about 10% fewer recognition errors than doing N-best re-scoring using the identical language model. This demonstrates the importance of using the unified model in the early stage of speech decoding.

The spoken language understanding (SLU) engine used in MiPad is based on a robust chart parser [26] and a plan-based dialog manager [27], [28]. Each semantic object defined and used for SLU is either associated with a real-world entity or an action that the application takes on a real-entity. Each semantic object has slots that are linked with their corresponding CFG. In contrast to the sophisticated prompting response in voice-only conversational interface, the response is a direct graphic rendering of the semantic object on MiPad's display. After a semantic object got updated, the dialog manager fulfills the plan by executing application logic and error repair strategy.

One of the critical tasks in SLU is semantic grammar authoring. It is necessary to collect a large amount of real data to enable the semantic grammar to yield a decent coverage. For spontaneous PIM application, MiPad SLU engine's slot parsing error rate in the general Tap and Talk field is above 40%. About half of these errors are due to the free-form text that are related to email or meeting subjects.

After collecting additional MiPad data, we are able to reduce the SLU parsing error by more than 25%, which might still be insufficient to be useful. Fortunately, with our imposed context constraints in the *Tap and Talk* interface, where slot-specific language and semantic models can be leveraged, most of today's SLU technology limitations can be overcome.

V. MiPad USER INTERFACE DESIGN AND EVALUATION

As mentioned previously, MiPad does not employ speech synthesis as an output method. This design decision is motivated mainly by the following two reasons. First, despite the significant progress in synthesis technologies, especially in the area of concatenated waveforms, the quality of synthesized speech has remained unsatisfactory for large scale deployments. This is also evident as the majority of commercial telephony speech applications still rely heavily on pre-recorded speech, with synthesized speech playing a minor role. The most critical drawback of speech output, however, is perhaps not with the quality of synthesized speech, which hopefully can be further improved, but with the nonpersistent or *volatile* nature of speech presentation. The human user must process the speech message and memorize the contents of the message in real time. There is no known user interface design that can elegantly assist the human user for the cases where the speech waveform cannot be easily heard and understood, or there is simply too much information to be absorbed. In contrast, a graphical display can render a large amount of information persistently for leisure consumption, avoiding the aforementioned problems.

MiPad takes advantage of the graphical display in UI design. The graphical display simplifies dramatically the dialog management. For instance, MiPad is able to considerably streamline the confirmation and error repair strategy as all the inferred user intentions are confirmed *implicitly* on the screen. Whenever an error occurs, the user can correct it through the GUI or speech modalities that are appropriate and appear more natural to the user. Thanks to the display persistency, users are not obligated to correct errors immediately after they occur. The display also allows MiPad to confirm and ask the user many questions in a single turn. Perhaps the most interesting usage of the display, however, is the *Tap & Talk* interface.

A. *Tap & Talk Interface*

Because of MiPad's small form-factor, the present pen-based methods for getting text into a PDA (Graffiti, Jot, soft keyboard) are potential barriers to broad market acceptance. Speech is generally not as precise as a mouse or a pen to perform position-related operations. Speech interaction can also be adversely affected by unexpected ambient noise, despite the use of denoising algorithms in MiPad. Moreover, speech interaction could be ambiguous without appropriate context information. Despite these disadvantages, speech communication is not only natural but also provides a powerful complementary modality to enhance the pen-based interface if the strengths of using speech can be appropriately leveraged and the technology limitations be overcome. In Table II, we elaborate several cases which show that pen and speech can be complementary and used effectively for handheld devices. The advantage of pen is typically the weakness of speech and vice versa.

Through usability studies, we also observed that users tend to use speech to enter data and pen for corrections and pointing. Three examples in Table III illustrate that MiPad's *Tap and Talk* interface can offer a number of benefits. MiPad has a field that is always present on the screen as illustrated in MiPad's start page in Fig. 7(a) (the bottom gray window is always on the screen).

Tap & Talk is a key feature of the MiPad's user interface design. The user can give commands by tapping the *Tap & Talk* field and talking to it. *Tap & Talk* avoids the speech detection problem that is critical to the noisy environments encountered in MiPad's deployments. The appointment form shown on MiPad's display is similar to the underlying semantic objects. By tapping on the attendees field in the calendar card shown in Fig. 7(b), for example, the semantic information related to potential attendees is used to constrain both CSR and SLU, leading to a significantly reduced error rate and dramatically improved throughput. This is because the perplexity is much smaller for each slot-dependent language and semantic model. In addition, *Tap & Talk* functions as a user-initiative dialog-state specification. The dialog focus that leads to the language model is entirely determined by the field tapped by the user. As a result, even though a user can navigate freely using the stylus in a pure GUI mode, there is no need for MiPad to include any special mechanism to handle spoken dialog focus and digression.

TABLE II
COMPLEMENTARY STRENGTHS OF PEN AND SPEECH AS INPUT MODALITIES

Pen	**Speech**
Direct manipulation	Hands/eyes free manipulation
Simple actions	Complex actions
Visual feedback	No Visual feedback
No reference ambiguity	Reference ambiguity

B. *Visual Feedback for Speech Inputs*

Processing latency is a well recognized issue in user interface design. This is even more so for MiPad in which distributed speech recognition is employed. In addition to the recognition process itself, the wireless network further introduces more latency that sometimes is not easily controllable. Conventional wisdom for UI design dictates that filling the time with visual feedback not only significantly improves the usability, but also prevents users from adversely intervening an ongoing process that cannot be easily recoverable. For these reasons, MiPad adopts a visual feedback for speech inputs. In addition, we have designed the visual feedback to help the user avoid a common cause for recognition error—waveform clipping. As the user speaks, MiPad displays a running graphical volume meter reflecting the loudness of the recorded speech right beneath the input field being dictated to. When the utterance is beyond the normal dynamic range, red bars are shown to instruct the user to lower the voice volume. When MiPad detects the end of a user utterance and sends the speech feature to the host computer for processing, a progress bar is overlaid on top of the volume meter. Although the underlying speech application program interface (SAPI) can raise an event whenever the recognizer exits a word node on the grammar, we found channeling back this event to MiPad consumes too much network traffic, which seems to outweigh the benefits of a detail and precise progress report. As a result, the current implementation employs a best attempt estimate on the recognition and understanding progress, not unlike the progress bar commonly seen in a Web browser. The progress estimation is computed solely on the client side with no network traffic involved. Before the outcome is served back to MiPad, the user can click a cancel button next to the status bar to stop the processing at the host computer. If the status bar vanishes without changing the display, it indicates that the utterance has been rejected either by the recognizer or by the understanding system. MiPad's error repair strategy is entirely user initiative: the user can decide to try again or do something else.

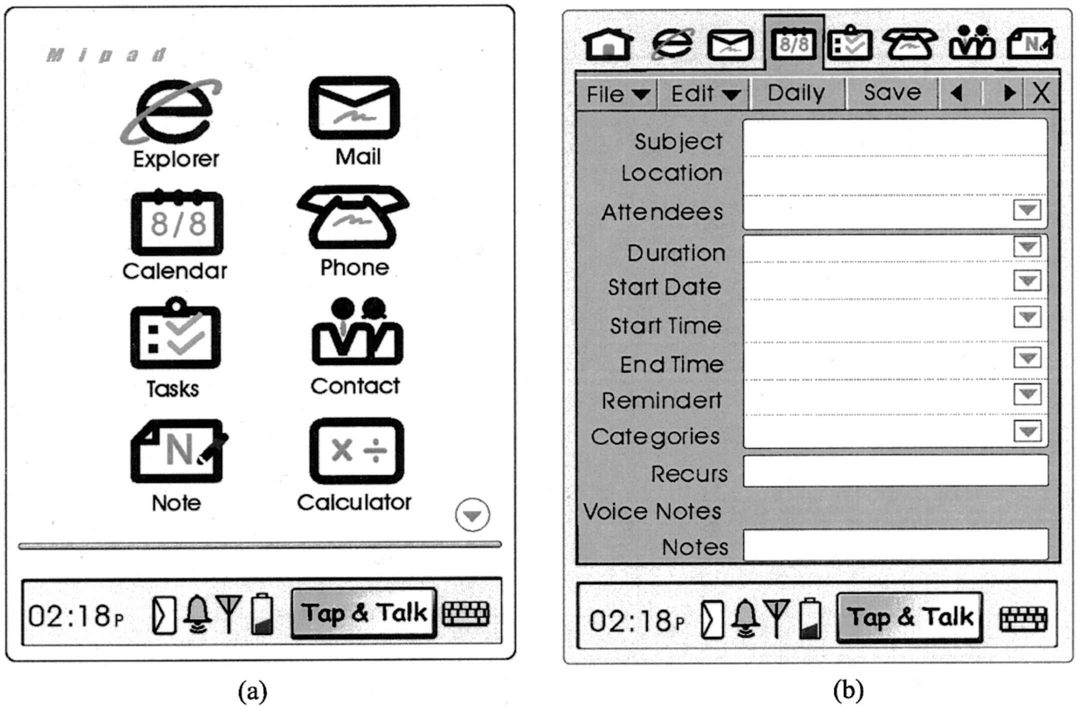

Fig. 7. Concept design for (a) MiPad's first card and (b) MiPad's calendar card.

TABLE III
THREE EXAMPLES SHOWING BENEFITS TO COMBINE SPEECH AND PEN FOR MIPAD USER INTERFACE

Actions	Benefits
Ed uses MiPad to read an e-mail, which reminds him to schedule a meeting. Ed taps to activate microphone and says *Meet with Peter on Friday.*	Using speech, information can be accessed directly, even if not visible. Tap and talk also provides increased reliability for ASR.
Ed taps Time field and says *Noon to one thirty*	Field values can be easily changed using field-specific language models
Ed taps Subject field dictates and corrects the text about the purpose of the meeting.	Bulk text can be entered easily and faster.

C. User Study Results

Our ultimate goal is to make MiPad produce real value to users. It is necessary to have a rigorous evaluation to measure the usability of the prototype. Our major concerns are:

"Is the task completion time much better?" and
"Is it easier to get the job done?"

For our user studies, we set out to assess the performance of the current version of MiPad (with PIM features only) in terms of task-completion time, text throughput, and user satisfaction. In this evaluation, computer-savvy participants who had little experience with PDAs or speech recognition software used the partially implemented MiPad prototype. The tasks we evaluated include creating a new appointment and creating a new email. Each participant completed half the tasks using the tap and talk interface and half the tasks using the regular pen-only the iPad interface. The ordering of tap and talk and pen-only tasks was random but statistically balanced.

1) Is Task Completion Time Much Better?: Twenty subjects were included in the experiment to evaluate the tasks of creating a new email, and creating a new appointment. Task order was randomized. We alternated tasks for different user groups using either pen-only or Tap & Talk interfaces. The text throughput is calculated during e-mail paragraph transcription tasks. On average it took the participants 50 s to create a new appointment with the Tap & Talk interface and 70 s with the pen-only interface. This result is statistically significant with $t(15) = 3.29$, $p < 0.001$. Time savings were about 30%. For transcribing an email it took 2 min and 10 s with Tap & Talk and 4 min and 21 s with pen-only. This difference is also statistically significant, $t(15) = 8.17$, $p < 0.001$. These time savings were about 50%. Error correction for the Tap & Talk interface remains as one of the most unsatisfactory features. In our user studies, calendar access time using the Tap & Talk methods is about the same as pen-only methods, which suggests that pen-based interaction is suitable for simple tasks.

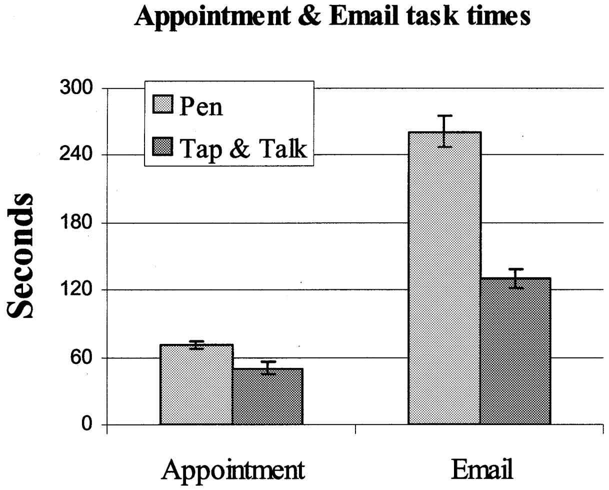

Fig. 8. User study on task completion times of email transcription and of making appointment, showing comparisons of the pen-only interface with the Tap and Talk interface. The standard deviation is shown above the bar of each performed task.

2) Is It Easier to Get The Job Done?: Fifteen out of the 16 participants in the evaluation stated that they preferred using the *Tap & Talk* interface for creating new appointments and all 16 said they preferred it for writing longer emails. The preference data is consistent with the task completion times. Error correction for the *Tap & Talk* interface remains as one of the most unsatisfactory features. On a seven-point Likert scale, with one being "disagree" and seven being "agree," participants responded with a 4.75 that it was easy to recover from mistakes.

Fig. 8 summarizes the quantitative user study results on task completion times of email transcription and of making appointment, showing comparisons of the pen-only interface with the *Tap & Talk* interface. The standard deviation is shown above the bar of each performed task.

VI. Summary

This paper describes work in progress in the development of a consistent human–computer interaction model and corresponding component technologies for multimodal applications. Our current applications comprise mainly PIM functions. Despite this incomplete implementation, we have observed that speech and pen have the potential to significantly improve user experience in our preliminary user study. Thanks to the multimodal interaction, MiPad also offers a far more compelling user experience than standard voice-only telephony interaction.

Though Moore's law also tells us that all the processing may be done in the device itself in the future, the success of the current MiPad depends on an always-on wireless connection. With upcoming 3G wireless deployments in sight, the critical challenge for MiPad remains the accuracy and efficiency of our spoken language systems since it is likely that MiPad will be used in noisy environments with no availability of a close-talking microphone, and the server also needs to support a large number of MiPad clients.

To meet this challenge, much of our recent work has focused on the noise-robustness and transmission efficiency aspects of the MiPad system. In this paper, we first described our new front-end speech processing algorithm development, based on the SPLICE technology, and some evaluation results. We then outlined some recent work on speech feature compression and error protection necessary to enable distributed speech recognition in MiPad. Various other MiPad system components, including user interface, HMM-based speech modeling, unified language model, and spoken language understanding, are also discussed. The remaining MiPad system components, i.e., dialog management, as well as its interaction with the spoken language understanding component, are not included in this paper; readers are referred to [27] and [28] for a detailed discussion of this topic.

Future development of spoken language systems in a mobile environment beyond MiPad will require us and the rest of the research community to face much greater challenges than we have encountered during the development of MiPad. One promising future direction for noise robustness which we will pursue includes intelligent combination of nonparametric approaches (such as SPLICE) and parametric approaches that take advantage of accurate knowledge of the physical nature of speech distortion. For example, knowledge of the phase relationship between the clean speech and the corrupting noise has been shown to be beneficial in providing better prior information in robust statistical feature extraction than the environment models which do not take account of the phase information [8], [13]. A combination of accurate acoustic environment models and knowledge about the speech distortion learned directly from stereo data will enable the recognizer's front end to effectively combat wider types and levels of speech distortion than our current algorithms can handle.

For future speech recognition technology to be usable in a mobile environment, it is necessary to break from the constrained vocabulary tasks as well as the relatively constrained speaker style. For example, in order to enable users to freely dictate e-mails, especially to friends and relatives, it may be difficult to constrain the vocabulary size and the strict dictation-like speaking style. More powerful speech recognition technology may be needed to achieve final success in such applications.

In the speech understanding and dialog management areas, we expect the multimodal integration in mobile environments to play a more dominant role than in the current MiPad. For example, the understanding component must be able to infer users' intention by integrating signals from a variety of input media. Cross-modality reference resolution becomes a key issue here. However, the increase in input modalities, together with a larger speech lexicon, will require understanding algorithms that deal more effectively with even higher perplexity. We anticipate that better dialog contextual management supplemented with external models of user preference will prove beneficial in successfully handling such a high-perplexity problem.

References

[1] A. Acero and R. Stern, "Robust speech recognition by normalization of the acoustic space," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Toronto, ON, Canada, 1991.

[2] M. Afify *et al.*, "Evaluating the Aurora connected digit recognition task: A Bell Labs approach," in *Proc. Eurospeech Conf.*, Aalborg, Denmark, Sept. 2001.

[3] L. Comerford, D. Frank, P. Gopalakrishnan, R. Gopinath, and J. Sedivy, "The IBM personal speech assistant," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, vol. 1, Salt Lake City, UT, May 2001.

[4] L. Deng, A. Acero, M. Plumpe, and X. D. Huang, "Large-vocabulary speech recognition under adverse acoustic environments," in *Proc. Int. Conf. on Spoken Language*, vol. 3, Beijing, China, Oct. 2000, pp. 806–809.

[5] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. D. Huang, "High-performance robust speech recognition using stereo training data," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Salt Lake City, UT, Apr. 2001.

[6] L. Deng, J. Droppo, and A. Acero, "Recursive estimation of nonstationary noise using a nonlinear model with iterative stochastic approximation," in *Proc. Automatic Speech Recognition and Understanding*, Dec. 2001.

[7] ——, "Robust speech recognition using iterative stochastic approximation and recursive EM for estimation of nonstationary noise," IEEE Trans. Speech Audio Processing, 2001, submitted for publication.

[8] ——, "Log-domain speech feature enhancement using sequential MAP noise estimation and a phase-sensitive model of the acoustic environment," in *Proc. Int. Conf. on Spoken Language Processing*, Denver, CO, Sep. 2002.

[9] De Veth *et al.*, "Feature vector selection to improve ASR robustness in noisy conditions," in *Proc. Eurospeech Conf.*, Aalborg, Denmark, Sept. 2001.

[10] J. Droppo, L. Deng, and A. Acero, "Efficient on-line acoustic environment estimation for FCDCN in a continuous speech recognition system," in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, Apr. 2001.

[11] ——, "Evaluation of SPLICE on the Aurora2 and Aurora3 tasks," in *Proc. Int. Conf. on Spoken Language Processing*, Denver, CO, Sept. 2002.

[12] ——, "Evaluation of the SPLICE algorithm on the Aurora2 database," in *Proc. Eurospeech Conf.*, Aalborg, Demark, Sept. 2001.

[13] J. Droppo, A. Acero, and L. Deng, "A nonlinear observation model for removing noise from corrupted speech log mel-spectral energies," in *Proc. Int. Conf. on Spoken Language*, Denver, CO, Sept. 2002.

[14] D. Ealey, H. Kelleher, and D. Pearce, "Harmonic tunneling: Tracking nonstationary noise during speech," in *Proc. Eurospeech Conf.*, Aalborg, Denmark, Sept. 2001.

[15] D. Ellis, W. Reyes, and M. Gomez, "Investigations into tandem acoustic modeling for the Aurora task," in *Proc. Eurospeech Conf.*, Aalborg, Denmark, Sept. 2001.

[16] R. Hamburgen, D. Wallach, M. Viredaz, L. Brakmo, C. Waldspurger, J. Bartlett, T. Mann, and K. Farkas, "Itsy: Stretch the bounds of mobile computing," *IEEE Computer*, pp. 28–36, Apr. 2001.

[17] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions ," in *ISCA ITRW ASR2000 "Automatic Speech Recognition: Challenges for the Next Millennium"*, Paris, France, Sept. 2000.

[18] X. D. Huang *et al.*, "MiPad: A next generation PDA prototype," in *Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, Oct. 2000.

[19] X. D. Huang *et al.*, "MiPad: A multimodal interaction prototype," in *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, vol. I, Salt Lake City, UT, Apr. 2001, pp. 9–12.

[20] D. Pearce, Ed., "ESE2 special sessions on noise robust recognition," in *Proc. Eurospeech Conf.* Aalborg, Denmark, Sept. 2001.

[21] R. Rose, S. Parthasarathy, B. Gajic, A. Rosenberg, and S. Narayanan, "On the implementation of ASR algorithm for hand-held wireless mobile devices," in *Proc. ICASSP*, vol. I, Salt Lake City, UT, Apr. 2001.

[22] J. Segura, A. Torre, M. Benitez, and A. Peinado, "Model-based compensation of the additive noise for continuous speech recognition: Experiments using the AURORA2 database and tasks," in *Proc. Eurospeech Conf.*, Aalborg, Denmark, Sept. 2001.

[23] O. Viikki, Ed., *Speech Communication (Special Issue on Noise Robust ASR)*, Apr. 2001, vol. 34.

[24] O. Viikki, I. Kiss, and J. Tian, "Speaker and language independent speech recognition in mobile communication systems," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Salt Lake City, UT, May 2001.

[25] Y. Wang, M. Mahajan, and X. Huang, "A unified context-free grammar and N-gram language model for spoken language processing," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, 2000.

[26] Y. Wang, "A robust parser for spoken language understanding," in *Proc. Eurospeech Conf.*, Budapest, Hungary, 1999.

[27] K. Wang, "Natural language enabled web applications," in *Proc. First NLP and XML Workshop*, Tokyo, Japan, Nov. 2001.

[28] ——, "Implementation of a multimodal dialog system using extended markup language," in *Proc. Int. Conf. on Spoken Language Processing*, Beijing, China, 2000.

Li Deng (S'83–M'86–SM'91) received the B.S. degree from the University of Science and Technology of China in 1982, the M.S. degree from the University of Wisconsin-Madison in 1984, and the Ph.D. degree in electrical engineering from the University of Wisconsin-Madison in 1986.

He worked on large vocabulary automatic speech recognition in Montreal, Canada, in 1986-1989. In 1989, he joined the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada as Assistent Professor; he became tenured Full Professor in 1996. From 1992 to 1993, he conducted sabbatical research at Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, and from 1997 to 1998, at ATR Interpreting Telecommunications Research Laboratories, Kyoto, Japan. In 1999, he joined Microsoft Research, Redmond, WA, as Senior Researcher, and is currently a principal investigator in the DARPA-EARS program and affiliate Professor of electrical engineering at University of Washington. His research interests include acoustic-phonetic modeling of speech, speech and speaker recognition, speech synthesis and enhancement, speech production and perception, auditory speech processing, noise robust speech processing, statistical methods and machine learning, nonlinear signal processing, spoken language systems, multimedia signal processing, and multimodal human-computer interaction. In these areas, he has published over 200 technical papers and book chapters, and has given keynote, tutorial, and other invited lectures worldwide. He recently completed the book *Speech Processing—A Dynamic and Optimization-Oriented Approach*.

Dr. Deng served on the Education Committee and Speech Processing Technical Committee of the IEEE Signal Processing Society during 1996-2000, and has, since Febuary 2002, been serving as Associate Editor for the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING.

Kuansan Wang received the B.S. degree from National Taiwan University in 1986, and the M.S. and Ph.D. degrees from the University of Maryland at College Park in 1989 and 1994, respectively, all in electrical engineering.

From 1994 to 1996, he was with the Speech Research Department at Bell Labs, Murray Hill, NJ. From 1996 to 1998, he was with speech and spoken language labs at NYNEX Science and Technology Center in White Plains, NY. Since 1988, he has been with speech technology group at Microsoft Research in Redmond WA. His research areas are speech recognition, spoken language understanding and multimodal dialog systems.

Alex Acero (S'83–M'90–SM'00) received an engineering degree from the Polytechnic University of Madrid, Spain, in 1985, the M.S. degree from Rice University, Houston, TX, in 1987 and the Ph.D. degree from Carnegie Mellon University, Pittsburgh, PA, in 1990, all in electrical engineering.

He was a Senior Voice Engineer at Apple Computer (1990–1991) and Manager of the Speech Technology Group at Telefonica Investigacion y Desarrollo (1991–1993). He joined Microsoft Research, Redmond, WA, in 1994, where he is currently Manager of the Speech Group. He is also Affiliate Professor at the University of Washington, Seattle, WA. He is author of the books *Spoken Language Processing* (Upper Saddle River, NJ, Prentice-Hall, 2000) and *Acoustical and Environmental Robustness in Automatic Speech Recognition* (Boston, MA, Kluwer, 1993). He also has written chapters in three edited books, seven patents and over 50 other publications. His research interests include noise robustness, speech synthesis, signal processing, acoustic modeling, statistical language modeling, spoken language processing, speech-centric multimodal interfaces, and machine learning. He is associate editor of *Computer, Speech, and Language*.

Dr. Acero served in the IEEE Signal Processing Society's Speech Technical Committee as member (1996–2000) and chair (2000–2002). He was general co-chair of the 2001 IEEE Workshop on Automatic Speech Recognition and Understanding, sponsorship chair of the 1999 IEEE Workshop on Automatic Speech Recognition and Understanding, and publications chair of *ICASSP '98*.

Hsiao-Wuen Hon (M'92–SM'00) received the B.S. degree in electrical engineering from National Taiwan University and the M.S. and Ph.D. degrees in computer science from Carnegie Mellon University, Pittsburgh, PA.

He is an Architect in Speech.Net at Microsoft Corporation. Prior to his current position, he was a Senior Researcher at Microsoft Research and has been a key contributor of Microsoft's Whisper and Whistler technologies, which are the corner stone for Microsoft SAPI and SDK product family. Before joining Microsoft, he worked at Apple Computer, Inc., where he was a Principal Researcher and Technology Supervisor at Apple-ISS Research Center. He led the research and development for Apple's Chinese Dictation Kit, which received excellent reviews from many industrial publications and a handful of rewards, including Comdex Asia'96 Best Software Product medal, Comdex Asia'96 Best of the Best medal and Singapore National Technology award. While at CMU, he was the co-inventor of CMU SPHINX system on which many commercial speech recognition systems are based on, including Microsoft and Apple. Hsiao-Wuen is an international recognized speech technologist and has published more than 70 technical papers in various international journals and conferences. He authored (with X. D. Huang and A. Acero) a book titled *Spoken Language Processing*. He has also been serving as reviewer and chairs for many international conferences and journals. He holds nine U.S. patents and currently has nine pending patent applications.

He is an associate editor for the IEEE TRANSACTIONS ON SPEECH AND AUDIO PROCESSING.

Jasha Droppo (M'01) received the B.S. degree in electrical engineering (*cum laude*, honors), from Gonzaga University in 1994. He received the M.S. degree in electrical engineering and the Ph.D. degree in electrical engineering from the University of Washington under L. Atlas in 1996 and 2000, respectively.

At the University of Washington, he helped to develop and promote a discrete theory for time-frequency representations of audio signals, with a focus on speech recognition. He joined the Speech Technology Group at Microsoft Research in 2000. His academic interests include noise robustness and feature normalization for speech recognition, compression, and time-frequency signal representations.

Constantinos Boulis is pursuing the Ph.D. degree at the University of Washington, Seattle. He received the M.Sc. degree from the Computer Engineering Department of Technical University of Crete, Greece from where he also holds an undergraduate degree.

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

Ye-Yi Wang (M'99) received the B.Eng. and M.S. degree in computer science and engineering from Shanghai Jiao Tong University in 1985 and 1987, respectively. He received the M.S. degree in computational linguistics and the Ph.D. degree in language and information technology from Carnegie Mellon University, Pittsburgh, PA, in 1992 and 1998, respectively.

He is currently a Researcher with the Speech Technology Group at Microsoft Research. His research interests include spoken language understanding, language modeling, machine translation, and machine learning.

Derek Jacoby received the B.S. degree in psychology from Rice University, Houston, TX, in 1995.

After working as a Consumer Products Usability Engineer at Compaq, he joined Microsoft in 1996 to work on the Systems Management Server. After four years of program management on the Windows team, he joined Microsoft Research to work on MiPad and associated projects. He is currently developing user interface approaches to adding speech recognition to the next version of Windows.

Milind Mahajan received the B.Tech. degree in computer science and engineering from Indian Institute of Technology, Mumbai, in 1986. He received the M.S. degree in computer science from University of Southern California, Los Angeles, in 1988.

He is currently a Researcher in the Speech Technology Group of Microsoft Research. His research interests include language modeling and machine learning.

Ciprian Chelba received the Dipl.-Eng. degree from the Politehnica University, Bucharest, Romania, in 1993. In 2000, he received the Ph.D. degree from The Johns Hopkins University, Baltimore, MD, working in the Center for Language and Speech Processing Lab.

He joined the Speech Technology Group at Microsoft Research in 2000. His main research interests are in the areas of statistical speech and language processing, particularly in language modeling and information extraction from speech and text. More broadly, he is in interested in statistical modeling.

Xuedong D. Huang (M'89–SM'94–F'00) received the B.S. degree in computer sciences from Hunan University, the M.S. degree in computer sciences from Tsinghua University, and the Ph.D. degree in electrical engineering from University of Edinburgh.

As General Manager of Microsoft .NET Speech, he is responsible for the development of Microsoft's speech technologies, speech platform, and speech development tools. He is widely known for his pioneering work in the areas of spoken language processing. He and his team have created core technologies used in a number of Microsoft's products including both Office XP and Windows XP, and pioneered the industry-wide SALT initiatives. He joined Microsoft Research as a Senior Researcher to lead the formation of Microsoft's Speech Technology Group in 1993. Prior to joining Microsoft, he was on the faculty of Carnegie Mellon's School of Computer Sciences and directed the effort in developing CMU's Sphinx-II speech recognition system. He is an affiliate Professor of electrical engineering at University of Washington, and an adjunct professor of computer science at Hunan University. He has published more than 100 journal and conference papers and is a frequent keynote speaker in numerous industry conventions. He has co-authored two books: *Hidden Markov Models for Speech Recognition* (Edinburgh, U.K.: Edinburgh University Press, 1990) and *Spoken Language Processing* (Englewood Cliffs, NJ: Prentice-Hall, 2001).

Dr. Huang received the National Education Commission of China's 1987 Science and Technology Progress Award, the IEEE Signal Processing Society's 1992 Paper Award, and Allen Newell Research Excellence Medal.