Subband Likelihood-Maximizing Beamforming for Speech Recognition in Reverberant Environments

Michael L. Seltzer, *Member, IEEE*, and Richard M. Stern, *Member, IEEE*

***Abstract*—In this paper, we introduce Subband LIkelihood-MAximizing BEAMforming (S-LIMABEAM), a new microphone-array processing algorithm specifically designed for speech recognition applications. The proposed algorithm is an extension of the previously developed LIMABEAM array processing algorithm. Unlike most array processing algorithms which operate according to some waveform-level objective function, the goal of LIMABEAM is to find the set of array parameters that maximizes the likelihood of the correct recognition hypothesis. Optimizing the array parameters in this manner results in significant improvements in recognition accuracy over conventional array processing methods when speech is corrupted by additive noise and moderate levels of reverberation. Despite the success of the LIMABEAM algorithm in such environments, little improvement was achieved in highly reverberant environments. In such situations where the noise is highly correlated to the speech signal and the number of filter parameters to estimate is large, subband processing has been used to improve the performance of LMS-type adaptive filtering algorithms. We use subband processing principles to design a novel array processing architecture in which select groups of subbands are processed jointly to maximize the likelihood of the resulting speech recognition features, as measured by the recognizer itself. By creating a subband filtering architecture that explicitly accounts for the manner in which recognition features are computed, we can effectively apply the LIMABEAM framework to highly reverberant environments. By doing so, we are able to achieve improvements in word error rate of over 20% compared to conventional methods in highly reverberant environments.**

***Index Terms*—Adaptive beamforming, microphone array processing, speech recognition.**

I. Introduction

THE PERFORMANCE of automatic speech recognition systems has improved to the point where commercial applications have been deployed for some small tasks. However, the benefits of speech-driven interfaces have yet to be fully realized, due in large part to the significant degradation in performance these systems exhibit in real-world environments. Improving speech recognition performance has been especially difficult in so-called *distant-talking* applications, i.e., applications in which the use of a close-talking microphone is either impractical or undesirable, and the microphone must be placed at some distance from the user. As a result of the increased distance between the user and the microphone, the speech signal becomes more susceptible to distortion from additive noise and reverberation effects which severely degrade the performance of speech recognition systems.

In these situations, microphone arrays have been used to mitigate the effects of this distortion. The corrupt speech signal is recorded over multiple spatially separated channels which are then processed jointly in order to spatially filter the soundfield and produce a cleaner output waveform.

Because traditional beamforming methods do not successfully compensate for the negative effects of reverberation on the speech signal, much recent research has focused on this area. One obvious approach to dereverberation is to estimate and then invert the room impulse response. Miyoshi *et al.* have shown that if multiple channels are used and the room transfer functions of all channels are known *a priori*, the exact inverse is possible to obtain if the transfer functions have no common zeros [1]. However, concerns about the numerical stability and, hence, practicality of this method have been raised because of the extremely large matrix inversions required [2], [3].

Other researchers have taken a matched filter approach to dereverberation. In [4], Flanagan *et al.* measure the transfer function of the source-to-sensor room response for each microphone and then use a truncated time-reversed version of this estimate as a matched-filter for that source–sensor pair. The matched filters are used in a filter-and-sum manner to process the array signals. While the authors demonstrate that the matched-filter approach has theoretical benefits over conventional delay-and-sum beamforming in terms of the signal-to-noise ratio (SNR), the matched-filter approach provides minimal improvement in speech recognition accuracy over delay-and-sum processing [5].

Another class of algorithms attempts to exploit characteristics of the speech signal or room transfer functions to perform dereverberation. For example, in [6], the kurtosis of the linear prediction residual of the speech signal is maximized to perform dereverberation. While the authors reported significant dereverberation as measured by informal listening tests, little improvement in speech recognition performance was achieved [7]. In another approach, room transfer functions are decomposed into minimum phase and all-pass components, and these components are processed separately to remove the effects of reverberation [8]. However, even in simulated environments, there were significant implementation difficulties in applying this method to continuous speech signals.

Manuscript received February 1, 2005; revised December 6, 2005. This work was supported by the Space and Naval Warfare Systems Center, San Diego, CA, under Grant N66001-99-1-8905. The work of M. L. Seltzer was also supported by a Microsoft Research Graduate Fellowship. The content of the information in this publication does not necessarily reflect the position or the policy of the U.S. Government, and no official endorsement should be inferred. The associate editor coordinating the review of this paper and approving it for publication was Dr. Maurizio Omologo.

M. L. Seltzer is with the Speech Technology Group, Microsoft Research, Redmond, WA 98052 USA (e-mail: mseltzer@microsoft.com).

R. M. Stern is with the Department of Electrical and Computer Engineering and School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: rms@sc.cmu.edu).

Digital Object Identifier 10.1109/TASL.2006.872614

Blind source separation (BSS) can also be interpreted as a microphone array processing problem, e.g., [9]. In the general BSS framework, observed signals from multiple sensors are assumed to be the result of a combination of source signals and some unknown mixing matrix. Various methods of estimating this mixing matrix, and thus separating the individual source signals, have been proposed, e.g., [10]. In [11], an analysis of BSS for the separation of convolutive mixtures of speech was performed, and it was found that BSS is equivalent to conventional adaptive beamforming and, therefore, cannot produce significant dereverberation. More recently, Buchner *et al.* showed that with appropriate changes to the BSS cost function, improved source separation of convolutive mixtures can be obtained [12].

One thing that unites practically all microphone array processing techniques is that they have been designed for speech *enhancement*. For speech recognition applications, one of these techniques is applied to the array signals in order to generate a enhanced single-channel waveform. The waveform is then input to the speech recognition system for feature extraction and decoding. Implicitly, this approach makes the assumption that generating an enhanced speech waveform will necessarily result in improved speech recognition. Such an assumption considers a human listener and a speech recognizer equivalent and ignores the manner in which speech recognition systems operate.

Speech recognition systems are statistical pattern classifiers that operate on a set of features extracted from the waveform. The hypothesis generated by the recognizer is the word string that has the maximum likelihood (ML) of generating the observed sequence of features, as measured by the recognizer's statistical models. Thus, any processing technique can only be expected to improve speech recognition performance if it maximizes, or at least increases, the likelihood of the correct hypothesis over other competing hypotheses. In contrast, the objective function of conventional microphone array processing algorithms is defined according to some signal level criterion, e.g., maximizing the SNR, minimizing the signal waveform error, or improving the perceptual quality as judged by human listeners. We believe that this mismatch between the objective criteria used by the array processing algorithms and that of the speech recognizer is the fundamental reason why sophisticated array processing algorithms may fail to produce significant improvements in speech recognition accuracy over far simpler methods, such as delay-and-sum beamforming [13].

To remedy this problem, we previously proposed a novel array processing method called LIkelihood-MAximizing BEAMforming (LIMABEAM). The goal of LIMABEAM is to find the set of array parameters that maximizes the likelihood of the correct recognition hypothesis. This is accomplished by explicitly considering the manner in which speech recognition systems process incoming speech and using pertinent information from the recognition engine itself to optimize the parameters of a sample-domain filter-and-sum beamformer. Exploiting the information contained in the recognizer in this way allows us to find the array parameters that maximize the likelihood that the features extracted from the output of the array will generate the correct recognition hypothesis.

We further suggest that maximizing the likelihood of the features input to a speech recognizer can be considered a primitive model of the minimization of distortion in the *effective* signal that mediates auditory perception. While current feature extraction for speech recognition is at best only a very crude approximation to the complex processing by the human auditory system, we expect that the successes of our approaches will motivate further efforts toward the development of other signal processing schemes more directly based on optimization of signals as they are presented to the auditory system.

The LIMABEAM approach has several advantages over other array processing methods. Most important, LIMABEAM is able to exploit the vast amount of *a priori* information about speech present in a speech recognizer. Speech recognizers are typically trained on tens, hundreds, or even thousands of hours of speech. Thus, the speech recognizer is, in essence, a detailed prior statistical model of speech. LIMABEAM uses this model to ensure that signal components important for recognition accuracy are enhanced without undue emphasis on less important components. In contrast, most other array processing algorithms are largely indifferent to the characteristics of the input signal and ignore this information. An exception to this is [14], in which a likelihood maximizing beamformer in the cepstral domain is proposed which also uses the statistical models of a speech recognizer. In addition, unlike some classical adaptive beamforming methods, no assumptions about the interfering noise are made, e.g., that it is uncorrelated from the target speech signal [15]. Finally, the proposed approach requires no *a priori* knowledge of the room configuration, array geometry, or source-to-sensor room impulse responses.

Experiments performed showed that LIMABEAM results in significantly improved speech recognition performance compared to traditional array processing approaches in noisy environments with low to moderate reverberation [16], [17]. However, even though its objective function is significantly different from conventional adaptive filtering schemes, LIMABEAM is at its core a gradient-descent-based least-mean-square (LMS) type of algorithm. As a result, like all LMS algorithms, its rate of convergence suffers when the input signals are highly correlated and the filter length is long [15]. Unfortunately, both of these conditions are generally true in highly reverberant environments. In addition, as the number of parameters of the beamformer to be jointly optimized increases, a significant increase in the amount of adaptation data is required.

The use of a subband filtering has been proposed as a means of improving the performance of adaptive filtering algorithms plagued by these problems, for many applications including acoustic echo cancellation and microphone array processing e.g., [18]–[20]. In general, developing a subband processing implementation of a full-band adaptive filtering algorithm is fairly straightforward. The signal is divided into subbands and the processing normally performed on the full-band signal is simply performed on each of the subbands independently. However, in LIMABEAM, the objective function measures the likelihood of a sequence of feature vectors against a set of statistical models. As a result, it is decidedly nontrivial to incorporate subband processing into the LIMABEAM framework.

In this paper, we present a new microphone array processing algorithm called Subband LIkelihood-MAximizing BEAMforming (S-LIMABEAM). S-LIMABEAM uses a novel subband filtering architecture which explicitly considers how recognition features are computed [21]. We demonstrate that an approach which processes all subbands independently, as is typically done in subband filtering algorithms, is in fact suboptimal for speech recognition applications. Instead, we propose to optimize selected groups of subbands jointly. By incorporating the proposed subband filtering architecture into the LIMABEAM framework, we are able to achieve significant improvements in speech recognition accuracy in reverberant environments.

The remainder of the paper is organized as follows. In Section II, we review the LIMABEAM algorithm. In Section III, we discuss the use of subband filtering for microphone array processing. In Section IV, we describe the S-LIMABEAM algorithm in detail. The performance of the proposed algorithm is evaluated through a series of experiments performed on speech captured in a variety of environments in Section V. Finally, we present the conclusion in Section VI.

II. LIMABEAM Algorithm

In conventional array processing algorithms, array parameters are chosen to optimize the beampattern, minimize signal distortion, or suppress interferring signals. Objective criteria such as these focus on the notion of a *desired signal*. However, speech recognition is not a signal processing problem, but rather a pattern classification problem. Sound classes are modeled by probability distribution functions. The speech waveform is converted into a sequence of feature vectors and the recognizer then compares these vectors to the statistical class models. The output is a label corresponding to the sound class or sequence of sound classes that has the maximum likelihood of generating the observed feature vectors.

Therefore, in order to improve speech recognition accuracy, the likelihood of the correct sound class must be maximized, or at least increased relative to the other (incorrect) classes for a given input. To do so, both the manner in which information is input to the recognizer (the feature extraction process) and the manner in which these features are processed (the decoding process) must be explicitly considered.

Speech recognition systems operate by finding the word string $\boldsymbol{w}$ most likely to generate the observed sequence of feature vectors $\mathcal{Z} = \{z_1, z_2, \ldots, z_T\}$, as measured by the statistical models of the recognition system. When the speech is captured by a microphone array, the feature vectors are a function of both the incoming speech and the array processing parameters, which we represent as $\boldsymbol{\xi}$. Recognition hypotheses are generated according to Bayes optimal classification as

$$\hat{\boldsymbol{w}} = \underset{\boldsymbol{w}}{\operatorname{argmax}} P\left(\mathcal{Z}(\xi)|\boldsymbol{w}\right) P(\boldsymbol{w}) \tag{1}$$

where the dependence of the feature vectors $\mathcal{Z}$ on $\boldsymbol{\xi}$ is explicitly shown. The acoustic score $P(\mathcal{Z}|\boldsymbol{w})$ is computed using the statistical models of the recognizer, and the language score $P(\boldsymbol{w})$ is computed from a language model.

In LIMABEAM, the array parameters $\boldsymbol{\xi}$ are chosen to maximize the likelihood of the correct transcription of the utterance that was spoken. This increases the difference between the likelihood score of the correct transcription and the scores of competing incorrect hypotheses, and thus increases the probability that the correct transcription will be hypothesized.

For the time being, let us assume that the correct transcription of the utterance, which we notate as $\boldsymbol{w}_C$, is known. We can then maximize (1) for the array parameters $\boldsymbol{\xi}$. Because the transcription is assumed to be known *a priori*, the language score $P(\boldsymbol{w}_C)$ can be neglected. The ML estimate of the array parameters can now be defined as the vector that maximizes the acoustic log-likelihood of the given sequence of words. In this paper, we assume that for a speech recognizer based on hidden Markov models (HMMs), the likelihood of a given transcription can be largely represented by the single most likely HMM state sequence. If $\mathcal{S}_C$ represents the set of all possible state sequences through this HMM and $\boldsymbol{s}$ represents one such state sequence, then the ML estimate of $\boldsymbol{\xi}$ can be written as

$$\hat{\boldsymbol{\xi}} = \underset{\xi, \boldsymbol{s} \in \mathcal{S}_C}{\operatorname{argmax}} \left\{ \sum_i \log\left(P(z_i(\boldsymbol{\xi})|s_i\right) + \sum_i \log\left(P(s_i|s_{i-1}, \boldsymbol{w}_C\right) \right\}. \tag{2}$$

Thus, according to (2), maximizing the likelihood of the correct transcription requires a *joint optimization* of both the array parameters and the HMM state sequence. This joint optimization can be performed by alternately optimizing the state sequence and the array processing parameters in an iterative manner.

For a given a set of array parameters $\boldsymbol{\xi}$, the speech waveforms can be processed and the features vectors extracted. Using the feature vectors and the known transcription, the most likely state sequence can be easily determined using the Viterbi algorithm [22]. For a given state sequence, finding the optimal array parameters depends on the form of the HMM state distributions, the feature vectors being used, and the beamforming architecture. In [16], we presented a method for finding the optimal array parameters of a sample-domain filter-and-sum beamformer when log mel spectra or mel frequency cepstral coefficients (MFCCs) are used as the features and the HMM states are represented by mixtures of Gaussians. Because both the feature extraction process and the state probability computation introduce nonlinearities into the relationship between the array parameters and the likelihood computation, finding the optimal array parameters requires the use of iterative nonlinear optimization techniques, such as the method of conjugate gradients [23].

A. LIMABEAM in Practice

Thus far, we have assumed that the correct transcription of the utterance $\boldsymbol{w}_C$ is known. For more realistic scenarios in which the transcription is in fact unknown, we developed two different implementations of LIMABEAM. The first, called *Calibrated LIMABEAM*, is appropriate for situations in which the user's position and environment do not vary significantly over time [24]. In this method, the user reads an enrollment utterance with a known transcription. An estimate of the most

TABLE I
WER OBTAINED USING DELAY-AND-SUM BEAMFORMING AND CALIBRATED LIMABEAM FOR TWO MICROPHONE ARRAY CORPORA WITH DIFFERENT REVERBERATION TIMES

T_{60}	WER (%)		Relative
	Delay-and-sum	Calib LIMABEAM	Improvement
0.30 s	13.0	11.0	15.4%
0.47 s	59.0	58.3	1.2%

likely state sequence corresponding to the enrollment transcription is made via forced alignment using the features derived from the array signals. These features can be generated using an initial set of filters, such as from a previous calibration session or a simple delay-and-sum configuration. Using this estimated state sequence, the filter parameters are then optimized. This constitutes a single iteration. The state sequence and the resulting filter parameters can then be refined using additional iterations until the overall likelihood converges. At this point, the calibration process is complete. The resulting filters are then fixed and used to process future incoming speech to the array.

The second method, called *Unsupervised LIMABEAM*, is more appropriate for use in time-varying environments [25]. In Unsupervised LIMABEAM, the array parameters are optimized for each utterance individually on the basis of a hypothesized transcription generated from an initial estimate of the array parameters. Using this hypothesized transcription and the feature vectors generated by the initial array parameters, the most likely state sequence is estimated using Viterbi alignment as before. The filters are then optimized using the estimated state sequence, and a second pass of recognition is performed. As in the calibrated case, this process can be iterated until the likelihood converges.

B. LIMABEAM in Reverberant Environments

Both the Calibrated LIMABEAM and Unsupervised LIMABEAM algorithms have been shown to produce significant improvements in recognition accuracy over conventional array processing approaches in environments with low reverberation [16]. In environments in which the reverberation is more severe, however, the improvements over traditional beamforming methods were reduced [17]. As an example, Table I compares the word error rate (WER) obtained by using delay-and-sum beamforming and Calibrated LIMABEAM in two different environments, one with a 60-dB reverberation time (T_{60}) of 0.30 s and one with a reverberation time of 0.47 s. In the 0.3-s reverberation environment, 50-tap finite-impulse response (FIR) filters were optimized, while in the 0.47-s reverberation environment, the filter length was increased to 100 taps. In both cases, one utterance (11.7 s) of calibration data was used per speaker. As the table shows, the improvement over delay-and-sum beamforming in the second, more reverberant, environment is only marginal, as compared to that obtained in the first environment.

In an effort to improve the performance of LIMABEAM in these conditions, we first increased the length of the filters used in the beamformer. Using longer filters may help compensate for the longer room impulse responses typically associated with more reverberant environments. Fig. 1 shows the WER in the 0.47-s environment as a function of beamformer filter length. As the figure shows, increasing the filter length does not improve the performance. In fact, using 200 taps per filter results in significantly worse performance than delay-and-sum beamforming, while reducing the filter length to 50 taps results in slightly improved performance.

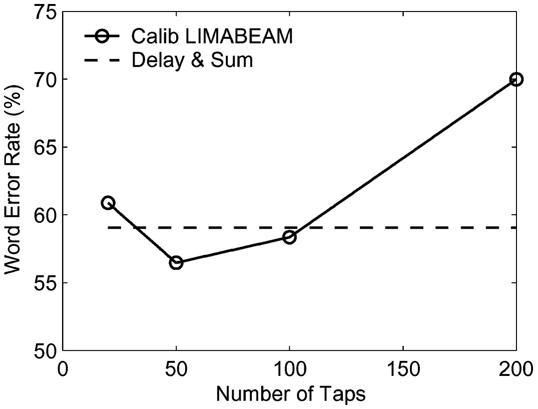


Fig. 1. WER as a function of filter length for the $\mathrm{WSJ}_{0.47}$ corpus when the filter parameters are optimized using Calibrated LIMABEAM. The performance using conventional delay-and-sum processing is also shown.

TABLE II
WER OBTAINED USING THE CALIBRATED LIMABEAM ALGORITHM FOR THE $\mathrm{WSJ}_{0.47}$ CORPUS WHEN 100-TAP FIR FILTERS ARE OPTIMIZED USING DIFFERENT AMOUNTS OF CALIBRATION DATA

# of Calib Utterance	Duration of Calib Data	WER (%)
1	11.7 s	58.3
2	18.9 s	53.5
3	28.1 s	57.0

As Fig. 1 suggests, if too many parameters are optimized using too short of a calibration utterance, overfitting can occur, and the resulting beamformer will not generalize well. Thus, we also attempted to improve performance of Calibrated LIMABEAM by increasing the amount of calibration data used for optimization. The results obtained when optimizing 100-tap FIR filters in the 0.47-s environment are shown in Table II.

As the table shows, increasing the amount of calibration data does not consistently improve the performance of Calibrated LIMABEAM, even though up to three times the amount of data are being used. One would certainly not expect the performance to degrade as the amount of calibration data is increased. However, using 28.1 s of data for calibration resulted in worse performance than using 18.9 s.

Thus, we were unable to improve the performance of Calibrated LIMABEAM significantly and consistently, either by increasing the number of beamformer parameters or the amount of calibration data used for optimization. Furthermore, to compensate for 0.47 s of reverberation, we expect that filters longer that 100 or 200 taps will be necessary. Yet, as we have shown here, even with this relatively modest number of taps, it is clear that it is difficult to find an optimal solution robustly. We hypothesize that as the number of parameters increases, the number of local minima in the optimization surface also increases, and thus, finding a robust solution becomes extremely difficult.

We note here that these experiments have been limited to Calibrated LIMABEAM because a more fundamental problem plagues Unsupervised LIMABEAM in highly reverberant environments. This problem has less to do with the effects of reverberation as it does with the nature of unsupervised optimization. For Unsupervised LIMABEAM to be successful, there must be a sufficient number of correctly labeled frames in the utterance. Performing unsupervised optimization on an utterance with too few correctly hypothesized labels will only degrade performance, propagating the recognition errors further. In these experiments, we typically use delay-and-sum beamforming as a means of obtaining first-pass hypothesized transcriptions. As Table I shows, in the environment with $T_{60} = 0.47$ s, the error rate of these hypotheses is almost 60%. At this level, little or no improvement can be expected using Unsupervised LIMABEAM unless more accurate first-pass transcriptions can be obtained. This general shortcoming of unsupervised processing does not change the previous conclusion that in reverberant conditions where long time-domain filters are required, finding optimal values of the LIMABEAM parameters has proven to be extremely difficult.

In the next two sections, we develop a LIMABEAM algorithm which utilizes subband processing techniques in order to improve the performance in these environments.

III. Subband Filtering for Microphone-Array-Based Speech Recognition

A. *Brief Review of Subband Adaptive Filtering*

The use of a subband filtering architecture has been proposed as a means to improve the rate of convergence of adaptive filtering algorithms when the desired filter to be estimated is very long and the input signals are highly correlated [15]. In subband filtering, the input signal is first decomposed into a series of independent subbands using a bank of bandpass filters, called *analysis filters*. Because each subband signal has a narrower bandwidth that the original signal, the signals can be downsampled. Each subband signal is now processed independently using an adaptive filter to minimize the *subband error*. After processing, the full-band signal is reconstructed by upsampling the output signals of the subband filters, and then passing them through another set of filters called *synthesis filters*.

Subband filtering provides an improvement in convergence over conventional full-band filtering for two reasons. First, when the signal is divided into subbands, the *learning rate* or *step size* used for adaptation in each subband can be chosen independently of the other subbands. By using subband-specific step sizes rather than a single step size for the entire broadband signal, it is possible to compensate for variations in the signal power across subbands and, as a result, obtain an improvement in convergence [15]. Second, because processing takes place in subbands, the number of parameters that needs to be estimated jointly is reduced. Because each subband filter is operating on a narrowband, downsampled version of the input signal, processing requires fewer parameters. This improves the computational complexity of the adaptation process. While the total computation can be shown to be approximately the same [26], the computation *per subband* is less. Because the subbands are independent, the adaptation of the different subband filters can be performed in parallel.

B. *Incorporating Subband Processing Into the ASR Front End*

In the feature extraction process used by most state-of-the-art speech recognition systems, the incoming waveform is first segmented into a series of overlapping frames. Each frame is then windowed and transformed to the frequency domain using a discrete Fourier transform (DFT). This short-time Fourier transform (STFT) generates a series of spectral vectors that reflect the change in the speech spectrum over time. Log mel spectra and then mel cepstra are then extracted from these vectors through a series of additional processing stages [27].

The STFT can be interpreted as a filtering operation, where the window function combined with the DFT operation creates a bank of bandpass filters centered at the DFT bin frequencies and having the impulse response of the window function [28]. Furthermore, because of the shift typically performed between successive frames, the front end is also performing downsampling of the input signal.

Thus, subband processing can be easily incorporated into the speech recognition front end because the required analysis processing, i.e., the bandpass filtering and downsampling, does not require any additional computation. In addition, because the STFT vectors are converted to feature vectors for decoding, there is no need to resynthesize the full-band signal after processing.

C. *Subband Filter-and-Sum Array Processing*

When subband processing is performed using a DFT filter bank, the subband signals are simply the DFT coefficients themselves. Consider a sequence of spectral vectors derived from several frames of speech waveform. The DFT coefficients at a particular frequency over all frames are a time–series of (complex) samples that describes the variation over time in the signal at that particular frequency. In this paper, each subband is assigned an FIR filter with complex tap values. Furthermore, because we are operating in a multichannel microphone array environment, we assign one such filter to each channel in the array. This leads to a *subband filter-and-sum* array processing architecture, which can be expressed as

$$Y_i[k] = \sum_{m=0}^{M-1} \sum_{p=0}^{P-1} H_p^{m*}[k] X_m^{i-p}[k] \tag{3}$$

where $X_m^i[k]$ is the value of the STFT in subband k of the signal captured by microphone m at frame i, $H_p^m[k]$ is the pth complex tap of the subband filter assigned to that microphone and subband, and * denotes complex conjugation.

In the next section, we present a method for optimizing the parameters of a subband filter-and-sum beamformer which specifically targets speech recognition performance.

IV. Subband Likelihood-Maximizing Beamforming

In this section, we present a subband filter-and-sum architecture derived directly from the manner in which recognition features are computed. We then present an algorithm for optimizing the subband filter parameters using the LIMABEAM framework.

A. *Feature-Based Subband Filtering*

In conventional subband adaptive filtering techniques, the filter coefficients $H_p^m[k]$ for particular subband k are adapted independently from the other subbands. However, closer examination of the feature extraction process used in speech recognition will reveal that, for our purposes, this is suboptimal.

To compute MFCC features, the mel spectrum is first derived from the STFT by computing the energy in a series of weighted overlapping frequency bands. Each component of the mel spectral vector is computed as a linear combination of the energy in a particular subset of DFT subbands. If we define M_i^l as the lth component of the mel spectrum of frame i and $V^l[k]$ as the value of the lth mel triangle applied to subband k, this can be expressed as

$$M_i^l = \sum_{k=l_-}^{l_+} V^l[k] Y_i[k] Y_i^*[k] \tag{4}$$

where l_- and l_+ are the DFT bins corresponding to the left and right edges of the lth mel filter, respectively. Outside of this range, the value of $V^l[k]$ is 0.

Substituting (3) into (4) clearly reveals that a given mel spectral component M_i^l is a function of the subband filter parameters of all microphones and *all subbands in the frequency range spanned by its mel filter*. Processing all subbands independently ignores this relationship. A more optimal approach would consider this set of filter coefficients *jointly* for each mel spectral component, and in the following section, we describe a method that does so.

B. *Maximum-Likelihood Estimation of Subband Filter Parameters*

As before, we will assume that maximizing the likelihood of a recognition hypothesis can be accomplished by maximizing the likelihood of the most likely HMM state sequence for that transcription. We further assume that the components of the feature vectors are independent. This is the same assumption used by the recognizer in modeling the HMM state output distributions as Gaussians with diagonal covariance matrices. Under this assumption, the likelihood of a given state sequence can be maximized by maximizing the likelihood of each component in the feature vector independently.

If we operate in the log mel spectral domain, each component of the feature vector is a function of only a subset of DFT subbands, as shown in (4). Therefore, to maximize the likelihood of a given vector component, we only need to optimize the parameters of the subband filters that are used to compute that component. Note that if we were to operate directly in the cepstral domain, we could not do this because each cepstral coefficient is a linear combination of *all* log mel spectral components and, therefore, a function of *all* subbands.[1]

We can now define $\boldsymbol{\xi}_l$ to be the vector of subband filter parameters required to generate the lth log mel spectral component. $\boldsymbol{\xi}_l$ is a complex vector of length $M \cdot P \cdot (l_+ - l_- + 1)$ covering all filter taps of all microphones for the group of subbands from which the lth mel spectral component is computed. The length of $\boldsymbol{\xi}_l$ varies depending on the number of subbands used to compute a particular mel spectral component.

For each dimension of the feature vector $l = \{0, \ldots, L-1\}$, we want to maximize the log likelihood of the given HMM state sequence with respect to $\boldsymbol{\xi}_l$, the vector of subband array parameters for that dimension. Thus, we perform L maximum likelihood optimizations of the form

$$\hat{\boldsymbol{\xi}}_l = \underset{\boldsymbol{\xi}_l}{\operatorname{argmax}} \sum_i \log \left(P\left(z_i^l(\boldsymbol{\xi}_l) | s_i \right) \right) \qquad l = \{0, \ldots, L-1\} \tag{5}$$

where $z_i^l(\boldsymbol{\xi}_l)$ is the lth component of the log mel spectrum at frame i, and s_i is the most likely HMM state at frame i.

Fig. 2 shows an example of this ML subband filter optimization for an array of two microphones, for the lth log mel spectral component which is composed of three DFT subbands.

C. *Optimizing the Subband Filter Parameters*

Because of both the nonlinear operations in the feature extraction process and the form of the state output distributions used by the HMMs, i.e., mixtures of Gaussians, (5) cannot be directly maximized with respect to $\boldsymbol{\xi}_l$. Therefore we use iterative nonlinear optimization methods. We employ conjugate gradient descent as our optimization method. In order to do so, we need to compute the gradient of (5) with respect to the corresponding set of array parameters $\boldsymbol{\xi}_l$.

1) Gaussian State Output Distributions: If the HMM state output distributions are assumed to be Gaussian, then the log-likelihood expression in (5) can be written as

$$\begin{aligned} \mathcal{L}(\boldsymbol{\xi}_l) &= \sum_i \log \left(P\left(z_i^l(\boldsymbol{\xi}_l) | s_i \right) \right) \\ &= \sum_i -\frac{1}{2} \frac{\left(z_i^l(\boldsymbol{\xi}_l) - \mu_i^l \right)^2}{\sigma_i^{l2}} + \kappa_i^l \end{aligned} \tag{6}$$

where μ_i^l and σ_i^{l2} are the mean and variance of the lth dimension of the Gaussian of state s_i and $\kappa_i^l = -0.5 \log(\sqrt{2\pi\sigma_i^{l2}})$. It can be shown that the gradient of (6) can be expressed as

$$\nabla_{\boldsymbol{\xi}_l} \mathcal{L}(\boldsymbol{\xi}_l) = -\sum_i \frac{\left(z_i^l(\boldsymbol{\xi}_l) - \mu_i^l \right)}{\sigma_i^{l2}} \frac{\partial z_i^l(\boldsymbol{\xi}_l)}{\partial \boldsymbol{\xi}_l} \tag{7}$$

where $\partial z_i^l(\boldsymbol{\xi}_l) / \partial \boldsymbol{\xi}_l$ is the gradient vector. The gradient vector is a complex vector with dimension that varies according to the

[1] In most speech recognition systems, the mel triangles do not actually span the *entire* frequency range. The lowest frequency is typically between 100 and 150 Hz, and the highest frequency depends on the sampling rate but is usually somewhat less than the Nyquist frequency.

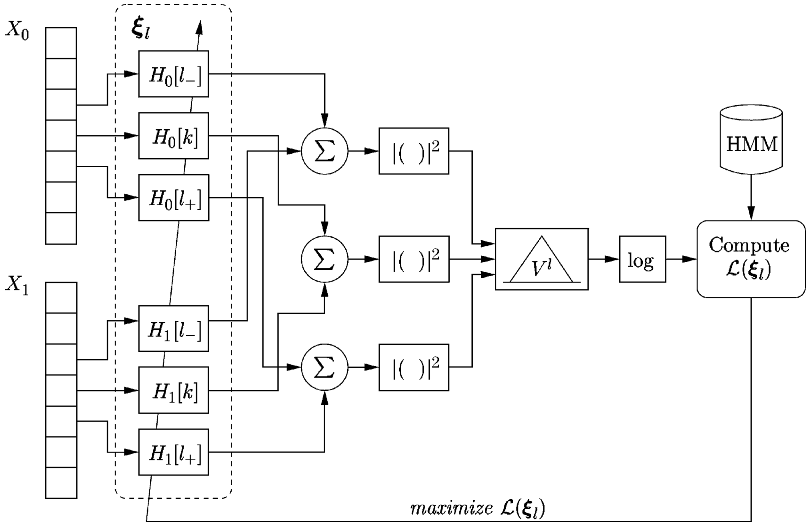


Fig. 2. S-LIMABEAM for an array of two microphones for the l log mel spectral component which is composed of three subbands. X_0 and X_1 are the STFT vectors for microphones 0 and 1, respectively, and V^l is the lth mel filter.

log mel spectral component. For the lth component, the length of the gradient vector is $M \cdot P \cdot (l_+ - l_- + 1)$. It can be shown that each element of the gradient vector can be expressed as

$$\frac{\partial z_i^l}{\partial H_p^m[k]} = 2\frac{V^l[k]}{M_i^l} X_{i-p}^m[k] Y_i^*[k] \tag{8}$$

where p is the tap index, m is the microphone index, and k is the subband index as before. The complete derivation of the gradient vector is given in the Appendix.

2) Mixture of Gaussians State Output Distributions: In the case where the state densities are mixtures of Gaussians, the gradient of the log-likelihood expression can be expressed as

$$\nabla_{\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{\xi}) = -\sum_{i} \sum_{k=1}^{K} \gamma_{ik}(\boldsymbol{\xi}_l) \frac{\left(z_i^l(\boldsymbol{\xi}_l) - \mu_{ik}^l\right)}{\sigma_{ik}^{l2}} \frac{\partial z_i^l(\boldsymbol{\xi}_l)}{\partial \boldsymbol{\xi}_l} \tag{9}$$

where $\gamma_{ik}(\boldsymbol{\xi}_l)$ is the *a posteriori* probability that the kth Gaussian in the mixture modeling state s_i generated the observed log mel spectral component $z_i^l(\boldsymbol{\xi}_l)$, and $\partial z_i^l(\boldsymbol{\xi}_l)/\partial \boldsymbol{\xi}_l$ is defined as in (8).

Because we are doing componentwise optimization, there are L separate optimizations performed, one for each dimension of the log mel spectral vector. Again, because we are performing subband processing, there are far fewer parameters to optimize *per optimization* than in the full-band case. Note, however, that because the mel triangles are typically spaced along the frequency axis so that adjacent triangles overlap each other by 50%, each DFT subband contributes to the value of two mel spectral components. By processing the DFT subbands jointly within each mel component, but independently across mel components, the optimization of the complete log mel spectral vector has twice as many degrees of freedom compared to conventional subband filtering schemes.

V. Experimental Results

To evaluate the performance of the proposed S-LIMABEAM algorithm, we performed a series of experiments on three different microphone array corpora representing a variety of acoustic environments.

The first corpus was created using the RWCP Soundscene Database [29]. This database contains room impulse responses recorded in five different rooms using linear and circular microphone arrays. The reverberation times of the rooms varied from 0.3 to 1.3 s. A reverberant corpus for speech recognition experiments was created by convolving the utterances from the WSJ0 test set [30] with the impulse responses recorded by a seven-element linear microphone array, with an intermicrophone spacing of 5.66 cm. The user was directly in front of the array at a distance of 2 m. A small amount of uncorrelated white noise was also added to each channel to simulate sensor noise. This corpus consists of five seperate test sets, each corresponding to a different reverberation time. We refer to these test sets as $\mathrm{WSJ}_{T_{60}}$, where T_{60} indicates the 60-dB reverberation time of the room [31]. For example, $\mathrm{WSJ}_{0.3}$ represents the test set from a room with a reverberation time of 0.3 s. Each test set consisted of eight speakers with approximately 40 utterances per speaker.

The second corpus used was the ICSI Meeting Recorder (ICSI-MR) corpus [32]. This corpus consists of recordings of actual meetings that took place over a three-year time period. The audio in each meeting was captured by a close-talking microphone worn by each user, as well as four pressure zone microphone (PZM) tabletop microphones placed along the conference room table and two microphones embedded in a wooden PDA mockup. The majority of the speech during these meetings was spontaneous multiparty conversation typical of meetings. In addition, during each meeting, each participant read several strings of connected digits.

Because the work in this paper is concerned with degradations in recognition accuracy caused by environmental conditions rather than speaking style, accent, or other factors, we chose to focus our experiments solely on the connected digits segments of the meetings. Furthermore, we restricted these data to only those meeting participants who were native speakers of English. The data set used for these experiments consisted of speech data from 16 different meetings, with an average of four people in each meeting. However, there were only 13 unique speakers in the data set, as some of the speakers participated in multiple meetings. The test set was 0.5 h in length.

In the experiments in this paper, we focused on improving the recognition accuracy using only the four PZM tabletop microphones. These microphones were spaced approximately 1 m apart along the center of the table. This microphone arrangement is highly suboptimal from a traditional beamforming point of view, as it produces severe spatial-aliasing over the range of frequencies spanned by speech signals.

Finally, to evaluate the performance of S-LIMABEAM in an environment with low reverberation but significant additive noise, experiments were performed using the CMU-8 corpus [33]. This database was recorded in the CMU speech lab. A linear microphone array of eight channels was used with an interelement spacing of 7 cm. The array was placed on a desk and user sat directly in front of the array at distance of 1 m. The reverberation time of the room was 0.24 s, and the speech captured by the array had an SNR of about 6.5 dB. The corpus consists of 140 utterances (10 speakers × 14 utterances). The utterances consist of strings of keywords as well as alphanumeric strings, where the user spelled out answers to various census questions, e.g., name, address, etc. This corpus was used to evaluate the original time-domain LIMABEAM algorithms extensively. [17].

Speech recognition was performed using Sphinx-3, a large-vocabulary HMM-based speech recognition system [34]. Context-dependent three-state left-to-right HMMs with no skips (eight Gaussians/state) were trained using the speaker-independent WSJ training set, consisting of 7000 utterances. The system was trained with 39-dimensional feature vectors consisting of 13-dimensional MFCC parameters, along with their delta and delta–delta parameters. A 25-ms window length and a 10-ms frame shift were used. Cepstral mean normalization (CMN) was performed in both training and testing.

Because the array parameter optimization of S-LIMABEAM is performed in the log mel spectral domain, but recognition is performed in the cepstral domain, we employ a second set of HMMs in the log mel spectral domain that are trained from the cepstral HMMs using the statistical reestimation (STAR) algorithm [35]. Training the log mel spectral models in this manner ensures that the two sets of models are exactly parallel, with identical frame-to-state alignments. This allows the decoding and Viterbi alignment to be performed in the cepstral domain and the array parameter optimization to be performed in the log mel spectral domain.

A. *Experimental Results Using Calibrated S-LIMABEAM*

To evaluate the performance of Calibrated S-LIMABEAM, we performed experiments using corpora captured in rooms with T_{60} of 0.3 and 0.47 s. For these experiments, a single iteration of calibration was performed as follows. Using the known transcription of the calibration utterance and features generated from the delay-and-sum output signal, the most likely state sequence was estimated. The filter parameters were then initialized to the delay-and-sum configuration and optimized. The state output distributions in the log-likelihood expression being maximized were represented by mixtures of eight Gaussians. Once the subband filter parameters were calibrated, they were used to process the remaining test set utterances. The same set of calibration utterances was used across all room conditions. Subband filters with one tap were used for the 0.3-s case, while filters with five taps were used for the 0.47-s case.

TABLE III
WER Obtained Using Delay-and-Sum Beamforming, Calibrated LIMABEAM, and Calibrated S-LIMABEAM for $\text{WSJ}_{0.3}$ and $\text{WSJ}_{0.47}$

T_{60}	WER (%)		
	Delay & Sum	Calibrated LIMABEAM	Calibrated S-LIMABEAM
0.30 s	13.0	11.0	9.8
0.47 s	59.0	56.5	41.7

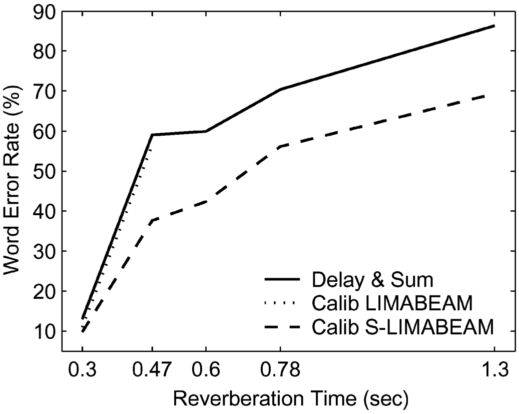


Fig. 3. WER obtained using Calibrated S-LIMABEAM shown as a function of reverberation time for the reverberant WSJ corpora. The performance of a delay-and-sum beamforming and the original full-band LIMABEAM algorithm are also shown.

The results of this experiment are shown in Table III. For comparison, the performance of delay-and-sum beamforming and full-band Calibrated LIMABEAM is also shown. In LIMABEAM, 50-tap FIR filters were optimized (shown to be the optimal filter length in Fig. 1). As the table shows, the performance of S-LIMABEAM is significantly better than both delay-and-sum beamforming and full-band LIMABEAM in both cases. The benefit in going from a full-band beamformer architecture to a subband architecture is particularly evident in the $\text{WSJ}_{0.47}$ case, where a 28.5% relative improvement over LIMABEAM is obtained by using S-LIMABEAM.

The performance of Calibrated S-LIMABEAM in environments with reverberation times up to 1.3 s is shown in Fig. 3. The performance of delay-and-sum beamforming is shown for comparison, as is the performance of LIMABEAM up to $T_{60} =$ 0.47 s. At longer reverberation times, the performance of full-band LIMABEAM is no better than delay-and-sum beamforming, and thus is not shown.

As the figure indicates, Calibrated S-LIMABEAM produces significant improvements over both conventional delay-and-sum processing and full-band LIMABEAM. Using this

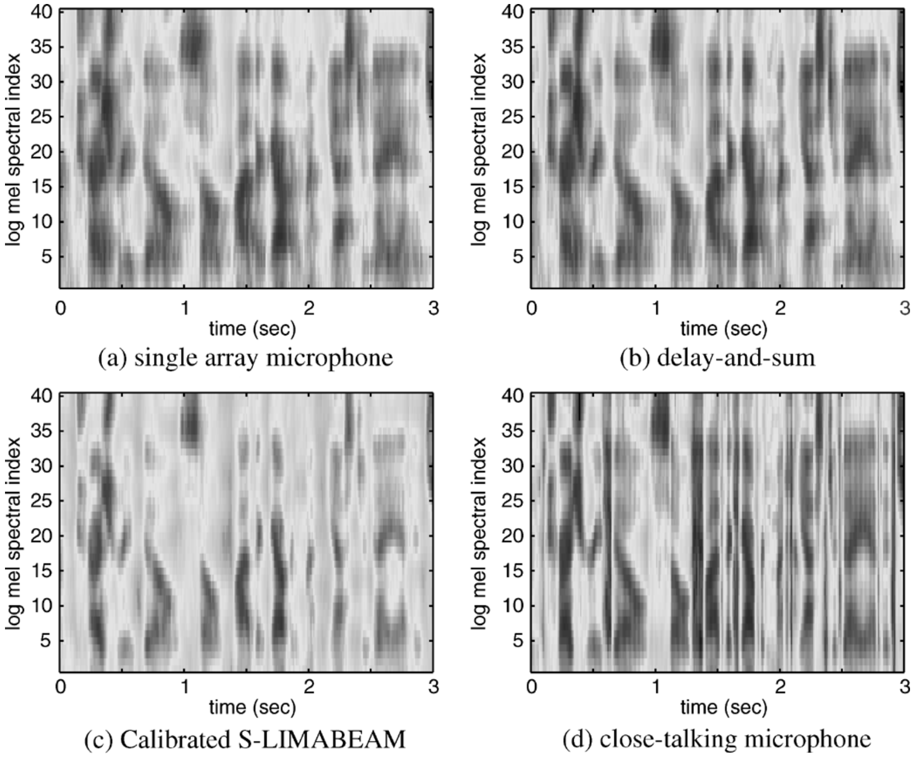


Fig. 4. Log mel spectrograms of a segment of an utterance from the $\mathrm{WSJ}_{0.47}$ corpus obtained from (a) a single channel in the array, (b) delay-and-sum beamforming, (c) the Calibrated S-LIMABEAM algorithm with five taps per filter, and (d) the close-talking microphone signal.

approach, the relative improvement over delay-and-sum beamforming, averaged over all reverberation times, is 26.0%, with a minimum improvement of 19.7% at 1.3 s and a maximum improvement of 36.2% at 0.47 s.

Fig. 4 shows four spectrographic displays of 40-dimensional log mel spectral feature vectors for a segment of one of the utterances in the test set. The figure compares the log mel spectra extracted from a single microphone from the array, the output of a delay-and-sum beamformer, the output of the Calibrated S-LIMABEAM algorithm with five taps per filter, and the close-talking recording. As the figure shows, delay-and-sum processing does little to reduce the temporal smearing caused by the reverberation, and in fact, the delay-and-sum spectrogram is virtually indistinguishable from that of the single microphone. Compared to the close-talking log mel spectra, all distinctions between high- and low-energy regions across time have been lost. On the other hand, the features generated by the calibrated subband filtering algorithm look significantly sharper and the low-energy regions between speech segments have been restored.

Clearly, we are able to achieve significant improvements in WER over a wide range of reverberation times. However, to be fair, we must also acknowledge that the data used in these experiments are ideally suited to a calibration algorithm. Because the reverberant speech corpora were created by convolving close-talking speech with recorded room impulse responses, the distortion caused by the reverberation was exactly the same for all utterances in the test set. This is a bit unrealistic, as even a user trying to remain in place would not be perfectly still. Therefore, it is possible that the algorithm's performance would degrade a bit if it were applied to data recorded by actual users. However, based on our results obtained from the original Calibrated LIMABEAM obtained using actual microphone array data, we expect the loss in performance to be minimal [17]. This hypothesis, however, remains untested, as a suitable reverberant corpus was not available.

These experiments show that the filter parameter calibration algorithm can be successfully incorporated into the S-LIMABEAM framework. We now turn to the unsupervised processing case for use in situations in which the environmental conditions and/or the user's position may vary across utterances.

B. Experimental Results Using Unsupervised S-LIMABEAM

To evaluate the performance of Unsupervised S-LIMABEAM, experiments were performed using the ICSI-MR corpus. We compared the recognition accuracies obtained using the single microphone with the highest SNR, and using all four microphones combined via delay-and-sum processing, Unsupervised LIMABEAM, and Unsupervised S-LIMABEAM. In order to choose the single best microphone, the SNR of each of the four microphones was estimated for every utterance using SNR estimation software from the National Institute of Standards and Technology (NIST) [36]. For each utterance, the microphone with the highest SNR was used for recognition.

For all utterances, a single iteration of Unsupervised LIMABEAM/S-LIMABEAM was performed as follows. Features derived from delay-and-sum processing were used to generate an initial transcription. Based on this transcription,

TABLE IV
WER OBTAINED ON THE ICSI-MR CORPUS USING ONLY THE FOUR PZM TABLETOP MICROPHONES. THE WER OBTAINED USING A CLOSE-TALKING MICROPHONE IS ALSO SHOWN FOR REFERENCE

Processing Method	WER (%)
Best Single Microphone	6.2
Delay-and-sum	2.7
Unsupervised LIMABEAM	2.6
Unsupervised S-LIMABEAM	2.2
Close-talking Microphone	1.1

TABLE V
WER OBTAINED USING THREE DIFFERENT ARRAY PROCESSING TECHNIQUES FOR THE ICSI-MR, WSJ$_{0.3}$, AND, WSJ$_{0.47}$ CORPORA

	WER (%)		
Corpus	Delay & Sum	Calib S-LIMABEAM	Unsuper S-LIMABEAM
ICSI-MR	2.7	2.6	2.2
WSJ$_{0.3}$	13.0	9.8	9.9
WSJ$_{0.47}$	59.0	41.7	53.9

the most likely HMM state sequence was estimated and used to optimize the beamformer. For full-band LIMABEAM, a beamformer with 50 taps per filter was optimized, while in S-LIMABEAM, filters with a single tap per subband were used. Each utterance was then processed by its optimized filters and a second pass of recognition was performed.

The results of this experiment are shown in Table IV. As the table shows, although the microphone arrangement is highly suboptimal, delay-and-sum processing is able to improve performance over the single best microphone. A small additional improvement over delay-and-sum beamforming is obtained by Unsupervised LIMABEAM. However, the best results are obtained using Unsupervised S-LIMABEAM, which provides a 18.5% relative improvement over delay-and-sum processing, and a 15.4% relative improvement over full-band LIMABEAM. We also performed Unsupervised S-LIMABEAM using two taps per subband, rather than one, but the performance declined to 2.4% WER. We believe that the degradation caused by longer filters occurred because the utterances were rather short, and there was not enough speech to optimize twice the number of beamformer parameters.

C. Comparison of Calibrated and Unsupervised S-LIMABEAM

We also compared the performance of the Calibrated and Unsupervised S-LIMABEAM algorithms directly using the ICSI-MR, WSJ$_{0.3}$, and WSJ$_{0.47}$ corpora. The results are shown in Table V. The peformance of delay-and-sum beamforming is also shown for comparison. The results in the table demonstrate the relative strengths and weaknesses of the two processing approaches.

Using the ICSI-MR corpus, Unsupervised S-LIMABEAM outperforms Calibrated S-LIMABEAM. This is not surprising, as in a meeting room environment, users tend to move around significantly. Thus, a beamformer calibrated to one particular utterance may not be accurate for future utterances. On the other hand, by optimizing the beamformer for each utterance individually, Unsupervised S-LIMABEAM can account for such user movement and achieve good performance.

TABLE VI
WER OBTAINED ON THE CMU-8 CORPUS USING DELAY-AND-SUM PROCESSING, UNSUPERVISED LIMABEAM, AND UNSUPERVISED S-LIMABEAM

Array Processing Method	WER (%)
Delay-and-sum	38.7
Unsupervised LIMABEAM, 20 taps	30.2
Unsupervised S-LIMABEAM, 1 tap	30.3

The performance obtained using the WSJ$_{0.3}$ corpus shows that if the speaker and environment are slowly varying or stationary, such as in front of a kiosk or desktop PC, *and* the first-pass transcriptions (in this case obtained from delay-and-sum processing) are reasonably accurate, we can expect the performance of the two S-LIMABEAM algorithms to be similar. Of course, the performance of the unsupervised optimization is critically dependent on the accuracy of the first-pass transcription. This is demonstrated by the performance of WSJ$_{0.47}$, where Calibrated S-LIMABEAM is able to obtain significant improvement over delay-and-sum processing, while the improvement from Unsupervised S-LIMABEAM is much smaller. The performance of Unsupervised S-LIMABEAM is hindered by the high WER of the first-pass delay-and-sum-based transcriptions.

D. S-LIMABEAM in Environments With Low Reverberation

In this paper, we have proposed a subband filtering approach to the LIMABEAM framework. The algorithms presented were designed specifically to improve the performance of speech recognition in highly reverberant environments. However, these algorithms will be significantly more valuable if they are in fact general solutions for many environments, rather than limited solely to use in environments where the distortion is caused primarily by significant reverberation, rather than other sources, such as additive noise.

In this series of experiments, we use the CMU-8 corpus to compare the performed obtained using Unsupervised LIMABEAM with a 20-tap filter-and-sum beamformer to that obtained using Unsupervised S-LIMABEAM with a single tap per subband filter. In [17], 20 taps was determined experimentally to produce the best recognition results using unsupervised processing in this environment. In both cases, the unsupervised filter optimization was performed based on hypothesized transcriptions from delay-and-sum processing. The results of these experiments are shown in Table VI. For comparison, the WER obtained from delay-and-sum processing is also shown.

The performance of LIMABEAM and S-LIMABEAM are virtually identical. In fact, there is no statistically significant difference between the two methods. Thus, S-LIMABEAM is as effective as the original sample-domain LIMABEAM approach in environments where the distortion is largely caused by additive noise and the reverberation is less severe.

VI. Summary and Conclusion

We previously proposed a new approach to microphone array processing called LIMABEAM. In LIMABEAM, the parameters of a sample-domain filter-and-sum beamformer are optimized in order to maximize the likelihood of the correct recognition hypothesis, as measured by the statistical models of the recognition engine itself. This method was shown to produce significant improvements in recognition accuracy compared to more traditional array processing techniques based on waveform-level objective criteria. However, in highly reverberant environments, where long filter lengths are required and the input signals are highly correlated, the performance of LIMABEAM degraded.

In this paper, we proposed a new algorithm called S-LIMABEAM, designed specifically to improve hands-free speech recognition in reverberant environments. S-LIMABEAM utilizes a novel subband filter-and-sum architecture which explicitly takes into account the feature extraction process used for recognition. Because each mel spectral component is derived from the energy in multiple subbands, the filters assigned to these subbands are optimized jointly for each mel spectral component. Thus, compared to conventional subband processing, S-LIMABEAM performs an independent likelihood maximization for each log mel spectral component, rather than for each individual subband.

Two implementations of S-LIMABEAM were presented. Using Calibrated S-LIMABEAM, an average relative improvement in WER of 26.0% over delay-and-sum processing was obtained in environments with reverberation times up to 1.3 s. In contrast, the relative improvement of LIMABEAM over delay-and-sum beamforming as less than 5% in these highly reverberant environments.

Using Unsupervised S-LIMABEAM on the ICSI Meeting Recorder corpus, we also demonstrated an improvement of over 20% in recognition accuracy in a situation in which the array geometry is suboptimal and is in fact, unknown *a priori*. Because S-LIMABEAM is a purely data-driven algorithm and makes no assumptions about array geometry or room configuration, we were able to obtain significant improvements under highly suboptimal recording conditions.

Finally, we showed that S-LIMABEAM is not only useful in environments corrupted by significant amounts of reverberation, but can in fact obtain good results in environments with low reverberation and significant additive noise. This generality makes S-LIMABEAM useful across a wide variety of environmental conditions. Of course, there are limitations to this algorithm. For example, Unsupervised S-LIMABEAM relies on the first-pass recognition as the basis of the parameter optimization. If the accuracy of this pass is extremely poor, then the parameters will be optimized based on inaccurate state sequences and will result in poor results.

In [17] and [16], we showed that additional improvement in recognition accuracy can be obtained by combining LIMABEAM/S-LIMABEAM with single-channel feature-space noise robustness techniques, e.g., CDCN [37] and HMM model adaptation techniques, e.g., MLLR [38]. However, we believe further improvement can be obtained by fully integrating the benefits of all of these methods into a single algorithm. Such an algorithm could potentially include both the introduction of an explicit noise model and a joint optimization over both the array parameters and the acoustic model parameters.

Appendix
Derivation of the S-LIMABEAM Gradient Vectors

In this Appendix, we derive the expression for the gradient vector required for S-LIMABEAM. In this algorithm, subband filters operating on the output of a DFT filterbank are optimized to maximize the likelihood of the resulting log mel spectra. The likelihood of each log mel spectral component is maximized independently. Therefore, for each log mel spectral component, we require the corresponding gradient vector, composed of the partial derivatives of that particular log mel spectral coefficient with respect the each of the filter parameters of its constituent subbands.

We define $\boldsymbol{z}_i$ to be the log mel spectral feature vector of length L for frame i. Recall that each mel spectral component is the energy in a particular frequency band defined by an associated mel filter. Thus, the lth log mel spectral component can be expressed as

$$\begin{aligned} z_i^l &= \log\left(M_i^l\right) && (10)\\ &= \log\left(\sum_{k=l_-}^{l_+} V^l[k] S_i^Y[k]\right) && (11)\\ &= \log\left(\sum_{k=l_-}^{l_+} V^l[k] Y_i[k] Y_i^*[k]\right) && (12)\end{aligned}$$

where $Y_i[k]$ is the DFT of waveform $y[n]$ at frame i, $S_i^Y[k]$ is the magnitude squared of $Y_i[k]$, and $V^l[k]$ is the coefficient of the lth mel filter in frequency bin k. Complex conjugation is denoted by *. The limits of summation l_- and l_+ represent the lowest and highest bins, respectively, in the frequency band defined by the lth mel filter.

In the subband array processing algorithm, $Y_i[k]$ generated as the output of a subband filter-and-sum operation, expressed as

$$Y_i[k] = \sum_{m=0}^{M-1} \sum_{p=0}^{P-1} H_p^{m*}[k] X_{i-p}^m[k] \qquad (13)$$

where $X_i^m[k]$ is the value of the STFT in subband k from microphone m at frame i, and $H_p^m[k]$ is the pth complex tap of the subband filter assigned to that microphone and subband.

We define $\boldsymbol{\xi}_l$ to be the vector of array parameters needed to compute the lth log mel spectral component. By substituting (13) into (12), it is apparent that $\boldsymbol{\xi}_l$ is is a complex vector of length $M \cdot P \cdot (l_+ - l_- + 1)$ composed of the subband filter parameters $\{H_p^m[k]\}$ for $m = \{0, \ldots, M-1\}, p = \{0, \ldots, P-1\}$, and $k = \{l_-, \ldots, l_+\}$.

We can now define the gradient as the vector composed of the partial derivatives of z_i^l with respect to each of the elements of $\boldsymbol{\xi}_l$. We express this as

$$\frac{\partial z_i^l}{\partial \boldsymbol{\xi}_l} = \left[\frac{\partial z_i^l}{\partial H_0^0[l_-]}, \frac{\partial z_i^l}{\partial H_0^1[l_-]}, \ldots, \frac{\partial z_i^l}{\partial H_{P-1}^{M-1}[l_+]} \right]^T. \quad (14)$$

A. Computing the Elements of the Gradient Vector

We define one element of the gradient vector, corresponding to microphone n, tap q, and subband r, as $H_q^n[r]$. From (10) and (11), we can express $\partial z_i^l / \partial H_q^n[r]$ as

$$\begin{aligned} \frac{\partial z_i^l}{\partial H_q^n[r]} &= \frac{1}{M_i^l} \frac{\partial M_i^l}{\partial H_q^n[r]} && (15) \\ &= \frac{1}{M_i^l} \frac{\partial M_i^l}{\partial S_i^Y[r]} \frac{\partial S_i^Y[r]}{\partial H_q^n[r]} && (16) \\ &= \frac{V^l[r]}{M_i^l} \frac{\partial S_i^Y[r]}{\partial H_q^n[r]}. && (17) \end{aligned}$$

To compute $\partial S_i^Y[r] / \partial H_q^n[r]$, we first define the filter parameter $H_q^n[r]$ simply as

$$H_q^n[r] = a + jb. \quad (18)$$

We can now define $\partial S_i^Y[r] / \partial H_q^n[r]$ as

$$\frac{\partial S_i^Y[r]}{\partial H_q^n[r]} = \frac{\partial S_i^Y[r]}{\partial a} + j \frac{\partial S_i^Y[r]}{\partial b}. \quad (19)$$

Using (12), (13), and (18), the partial derivative of $S_i^Y[r]$ with respect to a can be computed as

$$\begin{aligned} \frac{\partial S_i^Y[r]}{\partial a} &= \frac{\partial}{\partial a} \left(Y_i[r] Y_i^*[r] \right) \\ &= Y_i[r] \frac{\partial Y_i^*[r]}{\partial a} + \frac{\partial Y_i[r]}{\partial a} Y_i^*[r] \\ &= Y_i[r] X_{i-q}^{n*}[r] + X_{i-q}^n[r] Y_i^*[r] \\ &= 2\Re \left\{ X_{i-q}^n[r] Y_i^*[r] \right\}. && (20) \end{aligned}$$

We similarly obtain the partial derivative of $S_i^Y[r]$ with respect to b as

$$\begin{aligned} \frac{\partial S_i^Y[r]}{\partial b} &= Y_i[r] \frac{\partial Y_i^*[r]}{\partial b} + \frac{\partial Y_i[r]}{\partial b} Y_i^*[r] \\ &= Y_i[r] j X_{i-q}^{n*}[r] - j X_{i-q}^n[r] Y_i^*[r] \\ &= j \left(X_{i-q}^{n*}[r] Y_i[r] - X_{i-q}^n[r] Y_i^*[r] \right) \\ &= j \left(2j \left\{ \Im X_{i-q}^{n*}[r] Y_i[r] \right\} \right) \\ &= -2\Im \left\{ X_{i-q}^{n*}[r] Y_i[r] \right\}. && (21) \end{aligned}$$

Substituting (20) and (21) into (19), we obtain the final expression for $\partial S_i^Y[r] / \partial H_q^n[r]$

$$\begin{aligned} \frac{\partial S_i^Y[r]}{\partial H_q^n[r]} &= 2\Re \left\{ X_{i-q}^n[r] Y_i^*[r] \right\} - 2j\Im \left\{ X_{i-q}^{n*}[r] Y_i[r] \right\} \\ &= 2 \left(\Re \left\{ X_{i-q}^n[r] Y_i^*[r] \right\} - j\Im \left\{ X_{i-q}^{n*}[r] Y_i[r] \right\} \right) \\ &= 2 X_{i-q}^n[r] Y_i^*[r]. && (22) \end{aligned}$$

Finally, by substituting (22) into (17), we can express the element of the gradient vector corresponding to microphone n, tap q, and subband r as

$$\frac{\partial z_i^l}{\partial H_q^n[r]} = 2 \frac{V^l[r]}{M_i^l} X_{i-q}^n[r] Y_i^*[r]. \quad (23)$$

The full gradient vector $\partial z_i^l / \partial \boldsymbol{\xi}_l$ defined in (14) can now be computed by evaluating (23) over all microphones $n = \{0, \ldots, M-1\}$, taps $q = \{0, \ldots, P-1\}$, and subbands $r = \{l_-, \ldots, l_+\}$.

Acknowledgment

The authors would like to thank C. Wooters and the ICSI speech group for graciously providing the data from their Meeting Recorder corpus.

References

[1] M. Miyoshi and Y. Kaneda, "Inverse filtering of room acoustics," *IEEE Trans. Speech Audio Process.*, vol. 36, no. 2, pp. 145–152, Feb. 1988.

[2] W. Putnam, D. Rocchesso, and J. Smith, "A numerical investigation of the invertibility of room transfer functions," in *Proc. WASPAA*, Mohonk, NY, Oct. 1995, pp. 249–252.

[3] H. F. Silverman, W. R. Patterson, and J. L. Flanagan, "The huge microphone array (HMA)" Brown Univ., Providence, RI, May 1996, Tech. Rep..

[4] J. L. Flanagan, A. C. Surendran, and E. E. Jan, "Spatially selective sound capture for speech and audio processing," *Speech Commun.*, vol. 13, no. 1–2, pp. 207–222, Oct. 1993.

[5] B. Gillespie and L. E. Atlas, "Acoustic diversity for improved speech recognition in reverberant environments," in *Proc. ICASSP*, Orlando, FL, May 2002, vol. 1, pp. 557–560.

[6] B. Gillespie, H. Malvar, and D. Florencio, "Speech dereverberation via maximum kurtosis subband adaptive filtering," in *Proc. ICASSP*, Salt Lake City, UT, May 2001, vol. 6, pp. 3701–3704.

[7] H. A. Malvar, 2002, personal communication.

[8] Q.-G. Liu, B. Champagne, and P. Kabal, "A microphone array processing technique for speech enhancement in a reverberant space," *Speech Commun.*, vol. 18, pp. 317–334, Jun. 1996.

[9] S. Kurita, H. Saruwatari, S. Kajita, K. Takeda, and F, Itakura, "Evaluation of blind signal separation method using directivity pattern under reverberant conditions," in *Proc. ICASSP*, Istanbul, Turkey, 2000, pp. 3140–3143.

[10] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.

[11] S. Araki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech," *IEEE Trans. Speech Audio Process.*, vol. 11, no. 2, pp. 109–116, Mar. 2003.

[12] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of blind source separation algorithms for convolutive mixtures based on second-order statistics," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 1, pp. 120–134, Jan. 2005.

[13] J. Bitzer, K. U. Simmer, and K.-D. Kammeyer, "Theoretical noise reduction limits of the generalized sidelobe canceller (GSC) for speech enhancement," in *Proc. ICASSP*, Phoenix, AZ, May 1999, vol. 5, pp. 2965–2968.

[14] D. Raub, J. McDonough, and M. Wolfel, "A cepstral domain maximum likelihood beamformer for speech recognition," in *Proc. ICSLP*, Jeju Island, Korea, Oct. 2004.

[15] S. Haykin*, Adaptive Filter Theory*. Englewood Cliffs, NJ: Prentice-Hall, 2002.

[16] M. L. Seltzer, B. Raj, and R. M. Stern, "Likelihood maximizing beamforming for robust hands-free speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 5, pp. 489–498, Sep. 2004.

[17] M. L. Seltzer, "Microphone array processing for robust speech recognition," Ph.D. dissertation, Dept. Elect. Comput. Eng., Carnegie Mellon Univ., Pittsburgh, PA, Jul. 2003.

[18] A. Gilloire and M. Vetterli, "Adaptive filtering in subbands with critical sampling: analysis, experiments and application to acoustic echo cancellation," *IEEE Trans. Signal Process.*, vol. 40, no. 8, pp. 1862–1875, Aug. 1992.

[19] W. H. Neo and B. Farhang-Boroujeny, "Robust microphone arrays using subband adaptive filters," in *Proc. ICASSP*, Salt Lake City, UT, May 2001, vol. 6, pp. 3721–3724.

[20] W. Liu, S. Weiss, and L. Hanzo, "Subband adaptive generalized sidelobe canceller for broadband beamforming," in *Proc. IEEE Workshop on Stat. Sig. Proc.*, Singapore, Aug. 2001, pp. 591–594.

[21] M. L. Seltzer and R. M. Stern, "Subband parameter optimization of microphone arrays for speech recognition in reverberant environments," in *Proc. ICASSP*, Hong Kong, China, Apr. 2003, vol. 1, pp. 408–411.

[22] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inf. Theory*, vol. 13, no. 2, pp. 260–269, Apr. 1967.

[23] J. Nocedal and S. Wright*, Numerical Optimization*. New York: Springer, 1999.

[24] M. L. Seltzer and B. Raj, "Calibration of microphone arrays for improved speech recognition," in *Proc. Eurospeech*, Aalborg, Denmark, Sep. 2001, vol. 2, pp. 1005–1008.

[25] M. L. Seltzer, B. Raj, and R. M. Stem, "Speech recognizer-based microphone array processing for robust hands-free speech recognition," in *Proc. ICASSP*, Orlando, FL, May 2002, vol. 1, pp. 897–900.

[26] S. S. Pradhan and V. U. Reddy, "A new approach to subband adaptive filtering," *IEEE Trans. Signal Process.*, vol. 47, no. 3, pp. 655–664, Mar. 1999.

[27] S. Davis and P. Mermelstein, "Comparison of parametric representation for monosyllablic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-28, no. 4, pp. 357–366, Aug. 1980.

[28] S. H. Nawab and T. F. Quatieri*, Advanced Topics in Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1988, ch. Short-Time Fourier Transform, pp. 289–337.

[29] S. Nakamura, K. Hiyane, F. Asano, T. Nishiura, and T. Yamada, "Acoustical Sound Scene Database in Real Environments for Sound Scene Understanding and Hands-Free Speech Recognition," in *Proc. Int. Conf. Lang. Resources Evaluation*, Athens, Greece.

[30] D. B. Paul and J. M. Baker, "The design of the Wall Street Journal-based CSR corpus," in *Proc. ARPA Speech and Natural Language Workshop*, Harriman, NY, Feb. 1992, pp. 357–362.

[31] H. Kuttruff*, Room Acoustics*, 4th ed. London, U.K.: Spon, 2000.

[32] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus," in *Proc. ICASSP*, Hong Kong, China, Apr. 2003, vol. 1, pp. 364–367.

[33] T. M. Sullivan, "Multi-microphone correlation-based processing for robust speech recognition," Ph.D. dissertation, Carnegie Mellon Univ., Pittsburgh, PA, Aug. 1996.

[34] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler, R. Stern, and E. Thayer, "The 1996 Hub-4 Sphinx-3 system," in *Proc. DARPA Speech Recognition Workshop*, Feb. 1997, DARPA.

[35] P. Moreno, B. Raj, and R. M. Stern, "A unified approach for robust speech recognition," in *Proc. Eurospeech*, Madrid, Spain, Sep. 1995, vol. 1, pp. 481–485.

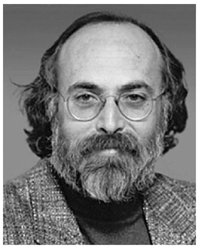
[36] D. S. Pallett, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, A. F. Martin, and M. A. Przybocki, "1995 HUB-3 NIST multiple microphone corpus benchmark tests," in *Proc. ARPA Speech Recognition Workshop*, Harriman, NY, Feb. 1996, pp. 27–46.

[37] A. Acero*, Acoustical and Environmental Robustness in Automatic Speech Recognition*. Boston, MA: Kluwer, 1993.

[38] C. J. Leggetter and P. C. Woodland, "Speaker adaptation of HMMs using linear regression" Cambridge University, Cambridge, U.K., Tech. Rep. CUED/F-INFENG/TR. 181, Jun. 1994.

Michael L. Seltzer (M'03) received the Sc.B. degree (with honors) from Brown University, Providence, RI, in 1996, and the M.S. and Ph.D. degrees from Carnegie Mellon University (CMU), Pittsburgh, PA, in 2000 and 2003, respectively, all in electrical engineering.

From 1996 to 1998, he was an Applications Engineer at Teradyne, Inc., Boston, MA, working on semiconductor test solutions for mixed-signal devices. From 1998 to 2003, he was a member of the Robust Speech Recognition group at CMU. In 2003, he joined the Speech Technology Group at Microsoft Research, Redmond, WA. His current research interests include speech recognition in adverse acoustical environments, acoustic modeling, microphone array processing, and machine learning for speech and audio applications.



Richard M. Stern (M'76) received the S.B. degree from the Massachusetts Institute of Technology (MIT), Cambridge, in 1970, the M.S. degree from the University of California, Berkeley, in 1972, and the Ph.D. degree from MIT in 1976, all in electrical engineering.

He has been a member of faculty of Carnegie Mellon University, Pittsburgh, PA, since 1977, where he is currently Professor of Electrical and Computer Engineering, and Professor by Courtesy of Computer Science, Language Technologies, and Biomedical Engineering. Much of his current research is in spoken language systems, where he is particularly concerned with the development of techniques with which automatic speech recognition systems can be made more robust with respect to changes of environment and acoustical ambience. He has also developed sentence parsing and speaker adaptation algorithms in earlier Carnegie Mellon speech systems. In addition to his work in speech recognition, he has also been active in research in psychoacoustics, where he is best known for theoretical work in binaural perception.

Dr. Stern has served on many technical and advisory committees for the DARPA program in spoken language research, and for the IEEE Signal Processing Society's technical committees on speech and audio processing. He was a corecipient of Carnegie Mellon's Allen Newell Medal for Research Excellence in 1992. He is a member of the Acoustical Society of America.