A Stochastic Segment Model for Phoneme-Based Continuous Speech Recognition

MARI OSTENDORF AND SALIM ROUKOS

Abstract—**Developing accurate and robust phonetic models for the different speech sounds is a major challenge for high-performance continuous speech recognition. In this paper, we introduce a new approach to modeling variable-duration phonemes, called the stochastic segment model. A phoneme X is observed as a variable-length sequence of frames where each frame is represented by a parameter vector and where the length of the sequence is random. The stochastic segment model consists of 1) a time warping of the variable-length segment X into a fixed-length segment Y called a resampled segment, and 2) a joint density function of the parameters of the resampled segment Y, which in this work is a Gaussian density. The segment model represents spectral/temporal structure over the entire phoneme. The model also allows the incorporation in Y of "acoustic-phonetic features" derived from X, in addition to the usual spectral features that have been used in hidden Markov modeling (HMM) and dynamic time warping approaches to speech recognition.**

In this paper, we describe the stochastic segment model, the recognition algorithm, and an iterative training algorithm for estimating segment models from continuous speech. We also present several results using segment models in two speaker-dependent recognition tasks and compare the performance of the stochastic segment model to the performance of hidden Markov models.

I. Introduction

In large vocabulary speech recognition, words are frequently modeled as networks of subword units such as phonemes. In other words, a word is modeled acoustically by concatenating phonetic acoustic models according to a pronunciation network stored in a dictionary of phonetic spellings. A benefit of this approach is that it is not necessary for the speaker to train all words in the vocabulary; only the phonetic models need to be trained. The goal of this work is to develop an improved approach to modeling the acoustics of phonetic units, in the context of a phoneme-based speech recognition system.

Hidden Markov modeling (HMM) is one method for probabilistic modeling of the acoustic realization of a phoneme. Although the HMM approach has been used successfully for modeling variable-duration phones [1]–[3], speech recognition performance is still far from perfect. We propose an alternative and novel approach, which we will refer to as the stochastic segment model, with the goal of improving speech recognition performance. The segment model represents the sequence of observation vectors over the entire speech segment of a phone. The motivation for looking at speech on a segmental level, rather than on a frame-by-frame basis, is that we can better capture the spectral/temporal structure over the duration of a phone. The usefulness of spectral correlation over the duration of a phonetic segment is evidenced in the success of segment-based vocoding systems [4], [5].

A speech "segment" is observed as a variable-length sequence of feature vectors where the features might be, for example, cepstral coefficients. The segment model is defined on a fixed-length representation of the variable-length observed segment, which is obtained by a time-warping (or resampling) transformation. The stochastic segment model uses a multivariate Gaussian density function to describe the resampled segment, and the recognition algorithm chooses the phoneme sequence according to a maximum *a posteriori* rule on the resampled segments. The Gaussian segment models must be estimated, or trained, from speech which has been segmented and resampled. One solution to the automatic training problem presented here is an iterative algorithm that first chooses the maximum probability phonetic segmentation and then uses a maximum likelihood density estimate from the resampled segments given by the segmentation. Using the stochastic segment model, initial experiments have demonstrated 74 percent phoneme recognition and 83 percent word recognition (350 word vocabulary, no grammar) for speaker-dependent, continuous speech recognition. These results compare favorably to discrete HMM performance using the same number of phoneme models on the same tasks: 62 percent phoneme recognition and 76 percent word recognition.

The approach we took to designing a segment-based recognition system involved several steps. First, we designed a phonetic recognition system using speech which had been manually segmented into phones. This system used hand-segmented data for both training and recognition. Next, we implemented an automatic recognition algorithm that jointly segments and recognizes phonemes, and we found that there was a small degradation in performance when phoneme segmentations are unknown for recognition. Again, the models for the automatic recognition experiments were trained from hand-segmented speech. Then we implemented a fully automatic training algorithm that estimates the phonetic models from contin-

Manuscript received January 28, 1987; revised December 19, 1988. This work was supported by the Advanced Research Projects Agency of the Department of Defense and was monitored by ONR under Contract N00014-85-C-0279.

M. Ostendorf is with the Department of Electrical, Computer, and Systems Engineering, Boston University, Boston, MA 02215.

S. Roukos is with IBM T. J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 8931334.

0096-3518/89/1200-1857$01.00 © 1989 IEEE

uous speech without requiring manually segmented data. There was no degradation in performance due to using automatic training rather than the hand-labeled speech. Finally, we incorporated the segment phoneme recognition system into a continous speech word recognition system.

Although we are strictly interested in phoneme recognition, we use several models per phoneme for some phonemes. For example, the three allophones for the phoneme "T"—flap (DX), unreleased "T" (URT) and released "T" (T)—are each modeled separately. We refer to the acoustic models of these allophones or phones as phonetic models. In this paper, we are not rigorous in distinguishing between the two terms phone and phoneme, which are used interchangeably. The context should allow the reader to disambiguate which unit we are describing. As a point of clarification, the observations and the models on the segment level represent *phones*. On the system level, the recognition performance is evaluated in terms of *phoneme* recognition.

The remainder of the paper is organized as follows. Section II introduces the segment model and discusses related work by others. Section III describes the segment-based recognition algorithm, and Section IV describes the training algorithm. In both Sections III and IV, we begin by describing each algorithm as implemented with presegmented speech, and then describe the corresponding extension of the algorithm to the case that includes automatic segmentation. Section V contains some experimental results for phoneme recognition in the context of system development. Section VI discusses the incorporation of the phonetic segment model in a word recognition system and presents results on a 350-word continuous speech recognition task. Since the best approach to continuous speech recognition has thus far been hidden Markov modeling, the recognition results of the segment-based systems are compared to the recognition results of HMM systems on the same tasks in both Sections V and VI. Section VII discusses the issue of complexity in a segment-based recognition system, and compares the segment-based system requirements to HMM system requirements. Finally, Section VIII contains a brief summary and suggestions for extensions of the model.

II. The Stochastic Segment Model

In this section, we describe the stochastic segment model, which consists of a resampling (time-warping) transformation and a multivariate Gaussian model for the resampled segment. We also mention some benefits of the segment structure, relating the different aspects of the segment model to other speech recognition approaches.

A phonetic model must account for both time and frequency structure. Most existing techniques, such as dynamic time warping [6] and hidden Markov modeling [1], model time-frequency by finding the best mapping of input frames to model states using a dynamic programming algorithm and by assuming independence over time for scoring spectral observations. The spectral scoring technique may be based on a minimum distance criterion [7] or on a maximum probability criterion, where probability can be represented using discrete distributions [1] or continuous distributions [8], [6]. Our goal is to better model the time dependence of spectral features in a phoneme. We accomplish this by using a constrained (but fixed) time-warping algorithm to give a fixed-length segment, as opposed to dynamic time warping. The fixed length segment is a sequence of spectral parameters which can be modeled jointly by a probability density function.

Observe a variable-length sequence of frames of speech $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_L]$ where $\boldsymbol{x}_i$ is a k-dimensional feature vector and L is the length of the segment. Given $\boldsymbol{X}$, we can find the fixed-length representation $\boldsymbol{Y} = \boldsymbol{X}T_L = [\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m]$ using a resampling or time-warping transformation T_L. T_L is an $L \times m$ matrix, so $\boldsymbol{Y}$ is an m-length sequence of k-dimensional vectors or a $k \times m$ matrix. $\boldsymbol{Y}$ can be thought of as the *underlying spectral trajectory* of $\boldsymbol{X}$, and $\boldsymbol{X}$ is the *realization* of $\boldsymbol{Y}$ with random length L due to variations in speaking rate. The phonetic segment model is based on the resampled segment $\boldsymbol{Y}$. Specifically, the segment model for each phone α is a conditional probability density of the resampled segment given that phone: $p(\boldsymbol{Y}|\alpha)$.[1] In this work, the density $p(\boldsymbol{Y}|\alpha)$ is assumed to be a multivariate Gaussian which represents the entire fixed-length segment, $\boldsymbol{Y}$, a km-dimensional model.

Resampling Transformations

The resampling transformation T_L is an $L \times m$ matrix used to transform an L-length observed segment $\boldsymbol{X}$ to an m-length *resampled* segment $\boldsymbol{Y}$. In the transformed segment $\boldsymbol{Y}$, each feature vector $\boldsymbol{y}_i$ is called a *sample* of the segment. Sampling, or "resampling," involves choosing m points in k-dimensional space which fall on the trajectory defined by the L observations which make up $\boldsymbol{X}$. We consider two approaches for variable- to fixed-length transformations: linear time sampling and space sampling.

Linear time sampling simply involves choosing m equally spaced times at which to resample the segment trajectory. With linear time sampling, the transformation T_L depends only on the observed segment length L. Space sampling involves choosing m sampling points which are equidistant (using Euclidean distance) along the segment trajectory in the k-dimensional feature space. With space sampling, T_L is a function of the observed segment $\boldsymbol{X}$. Fig. 1 illustrates linear time sampling and space sampling of a two-dimensional segment trajectory, such as the trajectory of the first and second cepstral coefficients. Note that fast transitions are sampled more densely by space sampling than by linear time sampling, causing fast transitions to be weighted more heavily in the representation $\boldsymbol{Y}$ when, in fact, the steady-state regions may represent a much larger proportion of the segment trajectory over time. To compensate for this effect, we compute a weight

[1]Since the submission of this paper, an alternative formulation was developed where the segment model is a probability density of the *observed* segment $\boldsymbol{X}$ (see [9]).

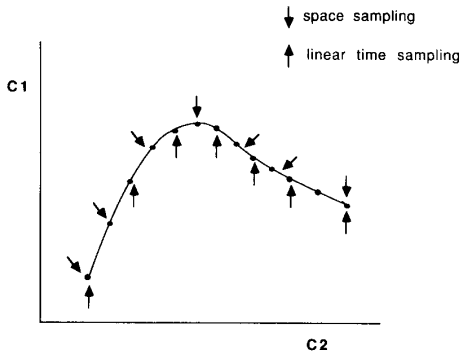
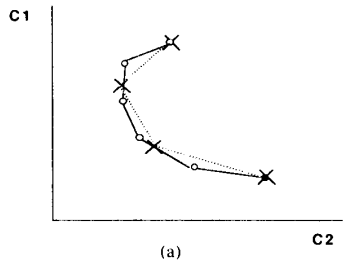


Fig. 1. Linear time sampling and space sampling of a two-dimensional trajectory. The circles represent actual observations. The arrows below the trajectory represent linear time-sampling points, and the arrows above the trajectory represent space-sampling points.



(a)

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & .33 & 0 & 0 \\ 0 & .66 & 0 & 0 \\ 0 & 0 & .66 & 0 \\ 0 & 0 & .33 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

(b)

Fig. 2. (a) A six-frame observed segment $\boldsymbol{X}$ is resampled using linear time sampling with interpolation, which results in a length-four resampled segment $\boldsymbol{Y} = \boldsymbol{X}T_6$. A circle (○) represents an observed frame, and a cross (+) represents a resampled segment value. (b) The transformation matrix T_6 for linear time-sampling with interpolation.

for each space sample, proportional to its duration in frames, for use in the pattern matching algorithm. Both the pattern matching algorithm and the weighting algorithm will be described further in Section III.

T_L can also be described by the m-length vector of sampling times $[t_1, \cdots, t_m]$. If a sample time is an integer, say $t_i = j$, then the sample value is the observation $\boldsymbol{x}_j$. If a sample time is noninteger, such as $t_i = 1.5$, then the sample value is found by interpolating between the two nearest observation vectors, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ in this case. Another option in choosing a transformation T_L involves sampling without interpolation, which simply means using the observation $\boldsymbol{x}_j$ closest in time to the sampling time. For example, using either space sampling or linear time sampling as described above, the resulting vector of sampling times is $[t_1, \cdots, t_m]$. Sampling without interpolation would yield the vector $[t_1', \cdots, t_m']$ where the new times are the nearest integer times $t_i' = \text{Int}(t_i)$ which correspond to actual observation times. A practical benefit of not interpolating between samples is that interpolation requires more computation than simply using the closest observation in time.

Both linear time sampling and space sampling are fixed warping algorithms in that they depend only on the observation $\boldsymbol{X}$, not on the phoneme model. An example of a warping algorithm which is dependent on both $\boldsymbol{X}$ and the model is dynamic time warping. We conjecture that a fixed warping which constrains the warping path over the phoneme is more robust for modeling spectral/temporal structure than dynamic time warping because the fixed warping does not allow extreme warpings that can result in recognition errors. However, we did not implement dynamic time warping to compare it to our choice of fixed-warping algorithms. Dynamic time warping within a segment is much more computationally expensive than the fixed warpings we have considered.

To illustrate the concepts of segments and resampling, consider a simple example using a two-dimensional feature space ($k = 2$ cepstral coefficients) and using $m = 4$ samples in the fixed-length model. Fig. 2(a) illustrates an observed phoneme segment $\boldsymbol{X}$ of length six and the resampled segment $\boldsymbol{Y}$. The circles in the figure denote the observed frames of $\boldsymbol{X}$, and the crosses denote the resampled values associated with linear time sampling using interpolation. The 6×4 ($L \times m$) linear sampling matrix is given in Fig. 2(b). (Had we sampled without interpolation, the nearest points to the arrows would have been chosen, and the corresponding T_6 matrix would have the same element values rounded to 1 or 0.) The probability of this segment given a phoneme α would be computed using the values in the trajectory of $\boldsymbol{Y}$.

Probabilistic Model

As we have mentioned, the segment model is a multivariate Gaussian based on the resampled segment $\boldsymbol{Y}$. The density $p(\boldsymbol{Y}|\alpha)$ or, equivalently, $\ln [p(\boldsymbol{Y}|\alpha)]$, can be thought of as a pattern match *score* for a segment $\boldsymbol{Y}$ and phoneme α. Recall that the resampled segments have km dimensions where k is the number of spectral features per sample and m is the number of samples. In this work, typically $k = 14$ and $m = 10$. Consequently, the segment model has 140 dimensions, ten 14-dimensional vectors of cepstral coefficients. In order to estimate the full 140-dimensional covariance matrix for each phone, we would need more than 140 observations of each phone to ensure that the covariance matrix is not singular. Currently, we do not have sufficient training data for this estimate, so we must make some simplifying assumptions about the structure of the model. For most experiments, we assume that the m samples in the resampled segment are independent of each other, which gives a block diagonal covariance structure where each block in the segment covariance matrix corresponds to a sample covariance. The log of the conditional probability of a segment $\boldsymbol{Y}$ given phoneme α can then be expressed as

$$\ln [p(\boldsymbol{Y}|\alpha)] = \sum_{j=1}^{m} \ln [p_j(\boldsymbol{y}_j|\alpha)] \qquad (1)$$

where $p_j(\mathbf{y}_j \mid \alpha)$ is a k-dimensional multivariate Gaussian model for the jth sample in the segment. $p_j(\mathbf{y}_j \mid \alpha) \sim N(\mu_j, C_j)$ where μ_j and C_j are, respectively, the mean and covariance of the jth sample and the notation $N(\mu, C)$ refers to a normal (Gaussian) density with mean μ and covariance C. $p_j(\mathbf{y}_j \mid \alpha)$ is estimated separately for each sample in each phoneme, so the density is both *sample-dependent* and *phone-dependent*. The block-diagonal structure also saves a factor of m in storage of covariance matrices and a factor on the order of m^2 in probability computation. In addition, in training the block diagonal structure, it is only necessary to invert m $k \times k$ matrices, instead of a full $km \times km$ matrix. The disadvantage of this approach is that the assumption of independence is not really valid. In particular, if the resampling method disallows interpolation, adjacent samples may be identical. In practice, making the independence assumption simply means not taking full advantage of the segment structure, which can model correlation between samples. In this case, the segment model is similar to a continuous HMM with a constrained state sequence.

The block-diagonal structure can also be used to model covariance in time instead of in frequency by rearranging the feature matrix $\mathbf{Y}$ by spectral coefficients rather than by samples. In this case, $\mathbf{y}_j$ would be an m-dimensional vector whose components are the values of the jth feature over the duration of the resampled segment. Using a block-diagonal covariance structure, there are k blocks, and $p_j(\mathbf{y}_j \mid \alpha)$ is the m-dimensional Gaussian model for the jth block. The recognition results for this structure were disappointing and are not reported in this work. One reason for this might be that the greater variability in time correlation may require a larger training set for reliable covariance estimates. Time correlation is being further investigated with a variation of this segment model which is described in [9].

It is likely that more detailed probabilistic models will yield better recognition results than the simple Gaussian model. For example, Gaussian mixture models [10] and context-dependent (conditional) models [2], [3] seem to be useful alternatives to simple phoneme models. However, we begin with a simple Gaussian phoneme-based model to prove feasibility of the segment structure. More detailed models will be discussed briefly at the conclusion of this paper.

Properties of the Segment Model and Related Work

There are several aspects of the stochastic segment model which we consider to be useful properties for a speech recognition system. These properties are time-warping constraints, the use of time-frequency correlation, a structure which allows segmental feature measurements, and an automatic training algorithm.

The transformation T_L, which maps the variable-length observation to a fixed-length segment, constrains the time warping over the phoneme duration. For example, a linear time-warping algorithm would result in a piecewise linear mapping between observations on model states, where the mapping is linear over the duration of a phoneme. The time-warping constraint prevents the possibility of any extreme warpings, such as mapping most of the acoustic observations to only one state of the phoneme, which could result in speech recognition errors. Kopec and Bush [7] also use a piecewise linear warping algorithm for these reasons. By constraining the time-warping algorithm, the model has a tighter duration constraint than that used in other models. For example, HMM's model the time warping by using state transition probabilities, allowing any warping (albeit low probability) [1]. Dynamic time-warping systems [6] typically use slope constraints to control the possible warpings, a local constraint that would still allow states of a phoneme model to be ignored. We believe that a constrained warping is a useful feature of the algorithm, although this claim is debatable.

Second, the segment model is a joint representation of the phoneme, so the model can capture correlation structure on a segmental level. In hidden Markov modeling, frames are assumed conditionally independent given the state sequence. With dynamic time warping, all observations are assumed independent. In the segment model, no assumptions of independence are necessary. However, in the work reported here, we do not take advantage of this aspect of the segment model since we use the assumption of sample independence [as given by (1)] because of limited training data in this study. We leave experimentation with time–frequency correlation in the segment model for future work. Makino and Kido [11] have reported results which model time–frequency correlation by using a segment-like structure for speech recognition referred to as the time-spectrum pattern (TSP). The full Gaussian covariance is used because the TSP has only 24 parameters, which are generated by taking the first five frames of a phoneme (a particular choice of the resampling transformation) and reducing the number of total spectral parameters using principle component analysis. A disadvantage of this model is that the whole phoneme is not included in the model.

By using a segment model, we can compute segment level features for phoneme recognition. In other words, since a segment spans an entire phoneme, the segment model provides a good structure for incorporating *segment-level* acoustic–phonetic features in a statistical (rather than rule-based) recognition system. By segment-level features, we mean features measured over the time span of a segment. For example, one might want to measure and incorporate the peak frequency of a formant or the average slope of a formant. Most other statistical recognition systems can incorporate only those acoustic–phonetic features which are measured on a frame-by-frame basis, such as short-time energy [12] and derivatives of spectral features [13], [14]. Since this work was submitted for publication, another system for incorporating segmental features has been proposed. Bush and Kopec [15] have reported results using several acoustic–phonetic features in their network-based recognizer, although the only useful segmental features were segment duration and peak

low-frequency energy. In the work presented here, we only investigate one segment-level feature: phoneme duration (Section V).

We point out that one important difference between the work reported here and that reported for many other models, excluding HMM's, is that we describe an automatic training algorithm with convergence properties and show that automatic training does not degrade performance relative to using hand-labeled training data. Kopec and Bush [7], [15] use a hand-segmented database for training initial models and bootstrap from these models to train models using new data. However, they make no claims as to properties of the bootstrapping technique nor do they show the effect of bootstrapping on recognition performance relative to hand labeling. Both Bocchieri and Doddington [6] and Makino and Kido [11] require hand alignment of tokens for training their models, although a bootstrapping algorithm could be developed for these techniques as well. Since there is a good possibility that performance will degrade in moving to automatic training algorithms, we feel it is important to demonstrate that our automatic training algorithm yields a system which performs as well as the system based on hand-segmented data.

III. Recognition Algorithm

In this section, we describe the algorithm for recognition using Gaussian segment models. We begin with the simple case of phoneme recognition in continuous speech using presegmented input data. Next, we generalize the algorithm to automatic recognition, that is, joint segmentation and recognition of the input speech.

The goal of the recognition algorithm is to maximize the recognition rate, which is the probability that the recognized phoneme $\hat{\alpha}$ (or word) equals the true phoneme α (or word). To accomplish this, we choose the phoneme sequence which maximizes the probability of the resampled segments, the maximum *a posteriori* (MAP) probability phoneme sequence. More specifically, the observed segments are first transformed to resampled segments, and the MAP probability phoneme sequence is found by computing probabilities of the resampled segments according to the collection of phonetic models $\{p(\boldsymbol{Y}|\alpha)\}$.

Known Segmentation

Begin by assuming that we are given a segment $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L]$, which is transformed to the resampled segment $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{T}_L = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m]$. The recognition algorithm is then a MAP rule given the resampled segment $\boldsymbol{Y}$:

$$\hat{\alpha} = \arg\max_{\alpha} p(\alpha|\boldsymbol{Y})$$

or, equivalently, using Bayes' rule and the fact that the segmentation is known,

$$\hat{\alpha} = \arg\max_{\alpha} \ln\left[p(\boldsymbol{Y}|\alpha)p(\alpha)\right] \tag{2}$$

where $\ln[p(\boldsymbol{Y}|\alpha)]$ is factored by samples (1) and can then be expressed as

$$\ln\left[p(\boldsymbol{Y}, \alpha)\right] = -(mn/2)\ln(2\pi) - 1/2\sum_{j=1}^{m}\left[(\boldsymbol{y}_i - \mu_j(\alpha))^T \cdot C_j(\alpha)^{-1}(\boldsymbol{y}_j - \mu_j(\alpha)) + \ln\left|C_j(\alpha)\right|\right] \tag{3}$$

where $\mu_j(\alpha)$ and $C_j(\alpha)$ are the mean and covariance of the jth sample of phoneme α and $|C|$ is the determinant of C. Maximizing $\ln[p(\boldsymbol{Y}, \alpha)]$ is equivalent to minimizing

$$D(\boldsymbol{Y}, \alpha) = \sum_{j=1}^{m}\left[(\boldsymbol{y}_j - \mu_j(\alpha))^T C_j(\alpha)^{-1} \cdot (\boldsymbol{y}_j - \mu_j(\alpha)) + \ln\left|C_j(\alpha)\right|\right] \tag{4}$$

which is the function actually implemented in the recognition system. All models are assumed to be phone-dependent unless otherwise stated.

One can think of $\ln[p(\boldsymbol{Y}|\alpha)]$ simply as a pattern match score, so other pattern matching algorithms can be used with the segment model. For example, the weighted Euclidean distance is another method of scoring segments which can be used in a minimum distance pattern match algorithm. The weighted distance measure for the segment model with m samples is

$$d(\boldsymbol{Y}, \alpha) = \sum_{j=1}^{m}(\boldsymbol{y}_j - \mu_j(\alpha))^T W_j(\alpha)(\boldsymbol{y}_j - \mu_j(\alpha)). \tag{5}$$

The Mahalanobis distance measure is one example of a weighted distance measure where the weight $W_j(\alpha)$ is equal to the inverse covariance of the distribution: $C_j(\alpha)^{-1}$. Observe that the Mahalanobis distance differs from the Gaussian log probability only by the determinant term. The minimum distance pattern matching approach is equivalent to a multisection VQ [16] using zero-rate codebooks. Note, however, that the Mahalanobis distance proposed here is phone-dependent, and the reported VQ codebook results do not use model-dependent distance measures.

Another example of a weighted distance measure would be weighting sample probabilities according to the sample duration, which is useful when using space-sampling transformations because space sampling emphasizes spectral transitions more than steady-state regions. In this case, the segment "score" is

$$d(\boldsymbol{Y}; \alpha) = \sum_{j=1}^{m} w_j \ln\left[p_j(\boldsymbol{y}_j|\alpha)\right] \tag{6}$$

where w_j is the weight corresponding to the duration in time of the jth sample in the resampled segment

$$w_j = 0.5(t_{j+1} - t_{j-1}) \tag{7}$$

using t_j as the sampling times defined in Section II and assuming $t_0 = 0$ and $t_{m+1} = L + 1$.

In order to discuss recognition of a sequence of phonemes, it is first necessary to introduce some notation. The underbar notation denotes a sequence. For example, $\underline{X} = \{X_i\}_{i=1,n}$ is a sequence of n observed segments and $\underline{Y} = \{Y_i\}_{i=1,n}$ is the corresponding sequence of resampled segments. A *segmentation* of a sequence of N input frames $\underline{x} = \{x_l\}_{l=1,N}$ is represented as $\underline{s} = \{s(i)\}_{i=1,n}$ where $s(i)$ is the index of the last frame in the ith segment. Using this notation, $X_i = [x_{s(i-1)+1}, \cdots, x_{s(i)}]$ where $s(0) = 0$. X_i is a segment of length $L(i) = s(i) - s(i-1)$ and $Y_i = X_i T_{L(i)}$.

The recognition algorithm for a sequence of phoneme segments with known segmentation is based on the assumption that consecutive phonemes are independent. Under the assumption of phoneme independence, each observed segment X_i is individually resampled to a fixed-length segment Y_i, which is used to find the MAP probability phoneme $\hat{\alpha}_i$. The recognized phoneme sequence is given by

$$\hat{\underline{\alpha}} = \arg\max_{\underline{\alpha}} \ln\left[p(\underline{Y}|\underline{\alpha})\, p(\underline{\alpha})\right] \tag{8}$$

where $\hat{\underline{\alpha}} = \{\hat{\alpha}_i\}_{i=1,n}$. Since we assume that the phonemes are independent, we have

$$\ln\left[p(\underline{Y}|\underline{\alpha})\right] = \sum_{i=1}^{n} \ln\left[p(Y_i|\alpha_i)p(\alpha_i)\right] \tag{9}$$

with the probability score given in (3). Note that this is equivalent to recognizing the segments independently using (2).

The minimum distance pattern match approach is analogous to (8), except that the probability score is replaced with a distance score (5) and "maximum" is replaced with "minimum."

Unknown Segmentation

In the automatic recognition system, we determine the segmentation of the input speech and recognize the phonemes. The Bayesian approach to finding the likelihood of a phoneme sequence is given by

$$l_B(\underline{\alpha}) = \sum_{\underline{s}} p(\underline{Y}(\underline{s})|\underline{\alpha})p(\underline{\alpha}) \tag{10}$$

where the notation $\underline{Y}(\underline{s})$ is used to denote the dependence of the resampled segment sequence on the segmentation. Since this approach is computationally expensive, we instead use a maximum likelihood detection method when some signal parameters are unknown [17, p. 292]. We consider the segmentation of the input speech as an unknown parameter. For each hypothesized sequence of phonemes, we determine the most likely segmentation by maximizing the likelihood

$$l(\underline{\alpha}) = \max_{\underline{s}} \ln\left[p(\underline{Y}(\underline{s})|\underline{\alpha})p(\underline{\alpha})\right]. \tag{11}$$

Then, the recognized phoneme sequence is obtained by

$$\hat{\underline{\alpha}} = \arg\max_{\underline{\alpha}} l(\underline{\alpha}). \tag{12}$$

This process is equivalent to 1) hypothesizing all possible segmentations, 2) transforming the segments to resampled form, 3) finding the best phoneme sequence and corresponding likelihood for each hypothesized segmentation (8), and 4) choosing the maximum likelihood phoneme sequence out of this set. For practical reasons, (11) is not strictly implemented. We use a sum of weighted log segment probabilities for the phoneme pattern match score, which is described below.

Joint segmentation and recognition involve allowing a variable segment rate where the average segment rate is controlled by incurring a cost for each segment used in recognition. This cost controls the phoneme insertion rate and can be thought of as the log probability term which corresponds to the distribution of the number of phonemes per sentence. The segment recognition system requires an additional cost factor because, for segments with observed length $L > m$, not all input observations contribute to the segment score. This aspect of the system tends to favor longer segments, which will have a lower average score (higher probability) per input observation. To compensate for this factor, each segment score is weighted by the duration of that segment, which ensures that an L-length observation contributes a proportionately large score. The score for a segmentation to be maximized over all allowable segmentations is then

$$J(\underline{Y}|\underline{\alpha}) = \sum_{i=1}^{n} \left\{\ln\left[p(Y_i|\alpha_i)\right]L(i) + \ln\left[p(\alpha_i)\right] + C\right\} \tag{13}$$

where n is the number of segments, C is the cost per segment, $L(i)$ is the duration of the ith segment in frames, and Y_i is determined by $\underline{s}$. The parameter C is used to control the phone insertion rate and can be thought of as corresponding to the log probability of the phone rate (number of phonemes/utterance). Note that $J(\underline{Y}|\underline{\alpha})$ replaces $\ln[p(\underline{Y}(\underline{s})|\underline{\alpha})p(\underline{\alpha})]$. Although this score is not strictly a likelihood, it can be thought of as an approximation of the probability of the L-long observation X which is based on the average sample score of Y. Note that using duration as a normalization factor in joint segmentation and recognition is different from using duration information as a feature in computing the probability of a sample. The duration weight adjusts the segment score independently of template duration, while the duration feature score is the conditional probability of the observed segment sample duration, given the template sample duration.

An efficient solution to (12) (modified to have weighted scores) is implemented using a dynamic programming algorithm. More specifically, at each time t, we compute the score for the best phoneme sequence ending at time t:

$$J_t^* = \max_{\tau,\alpha} \left\{J_\tau^* + \ln\left[p(Y(\tau, t)|\alpha)\right](t-\tau) + \ln\left[p(\alpha)\right] + C\right\} \tag{14}$$

where $\boldsymbol{Y}(\tau, t) = [\boldsymbol{X}_{\tau+1} \cdots \boldsymbol{X}_t]\, T_{t-\tau}$ and J_t^* is the score of the best phoneme sequence for the best resampled segment sequence for the observations $\{\boldsymbol{x}_l\}_{l=1,t}$. The phoneme α^* and the previous phoneme ending time τ^* which maximize (14) are saved for determining the final recognized phoneme sequence. The solution at the end of a sentence, time t_f, is given by $J_{t_f}^*$. The complexity of the search is proportional to the product of the number of phonetic models and the number of allowable phone durations.

IV. Training Algorithm

In this section, we look at training the segment model. We begin by describing the algorithm for estimating the models from a given segmented phoneme sequence. Next, we describe an algorithm for finding phoneme segmentations, given phoneme transcriptions and segment models. Finally, we show that the iteration of these two steps converges to a locally optimal collection of segment phoneme models.

Parameter Estimation

Given segmented data, it is possible to estimate the phonetic segment models from the statistics of the resampled segments. The phoneme *a priori* probabilities are estimated from the relative frequency of the phonemes in the training sequence. Assuming independent samples, a phonetic model is given by the m independent k-dimensional sample models. Therefore, it is simply necessary to estimate the sample Gaussian model for each of the m samples in each phoneme model from the statistics of the resampled phoneme segment samples. The sample density $N(\mu_j, C_j)$ for the jth sample of the phoneme α is determined by the maximum likelihood estimate for the mean and covariance from the set of resampled training vectors that map to sample j of phoneme α: $\{\, \boldsymbol{y}_{i,j};\ \boldsymbol{Y}_i \in A_\alpha \}$, where A_α is the set of all training segments $\boldsymbol{Y}_i$ which are transcribed as phoneme α and $\boldsymbol{y}_{i,j}$ is the jth sample of the resampled segment $\boldsymbol{Y}_i$.

It may be the case that there are not enough observations for some phonemes to estimate even a k-dimensional phone-dependent covariance model. There are several approaches to parameter estimation with small amounts of training. The approaches we investigated all involved making assumptions about the structure of the covariance matrix, such as a diagonal covariance or a phone-independent covariance. The best experimental results were achieved with the following scheme. With several observations, we can estimate parameter variance reasonably well, even though we might not have a good estimate of covariance. In this case, we use a diagonal covariance model. When there are even fewer observations of the phoneme, it is necessary to use a phone-independent covariance. (The phone-independent covariance is sample-dependent, however.) Thresholds for the different covariance structures are determined empirically as a function of the dimension of the feature space k. The results reported here are based on a threshold of k using the phone-dependent diagonal covariance and $2k$ for using the phone-dependent block-diagonal covariance. For phones that are observed fewer than k times, a phone-independent covariance is used.

Automatic Segmentation

Given a collection of Gaussian segment models, it is possible to find a good phoneme segmentation of a sentence when the phonetic transcription of the sentence $\underline{\boldsymbol{\alpha}}$ is known. (Phoneme transcriptions can be generated automatically from word transcriptions given a word pronunciation dictionary.) The algorithm is simply a full search of all possible segmentations $\underline{s}$ of the observed spectral vectors $\underline{\boldsymbol{x}}$ for the maximum probability segmentation $\hat{\underline{s}}$ given the phoneme transcription $\underline{\boldsymbol{\alpha}}$.

$$\hat{\underline{s}} = \arg\max_{\underline{s}} l(\underline{s}) \tag{15}$$

where the likelihood of a segmentation is given by

$$\begin{aligned} l(\underline{s}) &= \ln\left[p(\underline{\boldsymbol{Y}}(\underline{s})\,|\,\underline{\boldsymbol{\alpha}})\,p(\underline{\boldsymbol{\alpha}})\right] \\ &= \sum_{i=1}^{n} \ln\left[p(\boldsymbol{Y}_i\,|\,\alpha_i)\,p(\alpha_i)\right]. \end{aligned} \tag{16}$$

As in the recognition algorithm, we actually compute the sum of weighted log segment probabilities, so $J(\underline{\boldsymbol{Y}}\,|\,\underline{\boldsymbol{\alpha}})$ replaces $\ln[p(\underline{\boldsymbol{Y}}(\underline{s})\,|\,\underline{\boldsymbol{\alpha}})\,p(\underline{\boldsymbol{\alpha}})]$ (13).

An efficient solution to the maximization is based on an algorithm which is similar to that described in the previous section. Specifically, at each time t and for each possible i, we compute the score for the best i-length phoneme sequence ending at time t:

$$J_t^*(i) = \max_{\boldsymbol{Y}^i} J_t(\boldsymbol{Y}^i\,|\,\alpha^i) \tag{17}$$

where $J_t(\boldsymbol{Y}^i\,|\,\alpha^i)$ is the score of observations $\boldsymbol{Y}^i = [\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_i]$ given known phoneme sequence $\alpha^i = [\alpha_1, \cdots, \alpha_i]$. Using a dynamic programming solution, compute $J_t^*(i)$ by maximizing over the possible phoneme durations:

$$\begin{aligned} J_t^*(i) = \max_{\tau} \Big\{ & J_\tau^*(i-1) \\ & + \ln\left[p(\boldsymbol{Y}(\tau, t)\,|\,\alpha_i)\right](t-\tau) \Big\} \end{aligned} \tag{18}$$

where $\boldsymbol{Y}(\tau, t) = [\boldsymbol{X}_{\tau+1} \cdots \boldsymbol{X}_t]\, T_{t-\tau}$. Since the phoneme sequence is known in this case, the terms $\ln[p(\alpha_i)]$ and C in (13) do not affect the solution. The solution at the end of a sentence, time t_f, is given by $J_{t_f}^*(n)$. In automatic segmentation, the complexity of the search is proportional to the product of the number of segments in a sentence and the number of allowable phoneme durations.

Since the number of phonemes in the sequence is known, the duration weighting and insertion cost described in the previous section are not necessary; however, the duration weighted score (13) results in slightly better performance than the strict probability score in that the average segmentation error is smaller when compared to hand-segmented data.

Iterative Training Algorithm

Using the two steps described above, parameter estimation and automatic segmentation, we can define an iterative algorithm for automatically training the segment models from continuous speech.

Given:

- Phoneme transcription for training data.
- Initial Gaussian models for all phonemes: $\{p_0(\boldsymbol{Y}|\alpha)\}$.
- $t = 0$.

Iterate:

1) Find the maximum probability segmentation $\underline{\hat{s}}_t$ of the training data for the given transcriptions and the current probability densities $\{p_t(\boldsymbol{Y}|\alpha)\}$.

2) Find the maximum likelihood estimate for the densities $\{p_{t+1}(\boldsymbol{Y}|\alpha)\}$ using $\underline{\hat{s}}_t$.

3) $t <= t + 1$ and go to Step 1).

Observe that with each step, the likelihood of the segmentation $l(\underline{\hat{s}}_t)$ (16) increases for the given phoneme sequence. In Step 1), the new segmentation is the most likely segmentation for the current phoneme models, which must be at least as probable as the previous segmentation. In Step 2), the new densities increase the probability of the current segmentation by definition of the maximum likelihood estimate. If there are at least two observations of every phoneme in the transcription sequence and there are no two input observations with any identical features, then the probability of the transcription sequence is finite for any segmentation. There is a finite number of possible segmentations because the length of the training data is finite, so $l(\underline{\hat{s}}_t)$ is bounded by the likelihood of the best segmentation. Therefore, the segmentation likelihood sequence $l(\underline{\hat{s}}_t)$ converges to a local optimum. The sequence of segmentations converges to a local optimum when the segmentation likelihood sequence converges to a local optimum because there is a finite number of possible segmentations and the search for the maximum likelihood segmentation in Step 1) is ordered. Hence, the iterative training algorithm converges to a locally optimum collection of models.

V. Phoneme Recognition Results

Our approach to building a recognition system was: first, to test the algorithm and determine the model structure using hand-labeled data; second, to determine the performance with an automatic recognition algorithm; and finally, to demonstrate similar performance with an automatic training algorithm. In this section, we will present results for all three steps on a phoneme recognition task. These experiments represent the system development effort. The following section on word recognition describes the evaluation of the segment model in the context of a speech recognition system.

All experiments use $m = 10$ samples per segment and $k = 14$ cepstral coefficients per sample. These values are based on work in segment quantization [4], and limited experimentation with segment models for phoneme recognition confirmed that these values represent a reasonable compromise between complexity and performance for the segment-based recognition system as well as the quantization system. (Note that increasing the sample length improves performance or at worst does not degrade performance, but also increases complexity.) The possibility of using different numbers of samples in the models for different phonemes was not investigated in this work because we felt that it would increase the system complexity without providing performance improvement. When an observed segment is shorter than the model length, it is resampled by repeating the observed frames. Therefore, shorter phones will have models where the neighboring state distributions are quite similar. Information may be duplicated in the models, but there is no information lost and no lack of data for training the additional model states. Therefore, there is no advantage to using shorter length models when successive model samples are assumed independent. The idea of using variable-length models would be more useful in an alternative version of the stochastic segment model [9] which does not resample short phonemes by repeating observations.

We estimate the expected recognition rate by looking at the average recognition rate on an independent test set of continuous speech. We have 61 phonetic classes to disambiguate. However, in counting errors, different phones representing the same English phoneme can be substituted for each other without causing an error. For example, an "AX" (schwa) recognized as "IX" (fronted schwa) is considered acceptably correct, as is a "URT" (unreleased T) recognized as a "T." In this sense, all recognition rates presented represent "acceptably correct" recognition rates on the 61 different phonetic models used. The recognition rate is the correct recognition rate for the 42 phonemes we use for English. The phoneme recognition rate (42 phonemes) is typically 6–8 percent higher than the phonetic recognition rate (61 phones).

The database used to determine model parameters and structure is a single-speaker, continuous speech collection of 109 sentences, of which nine sentences are reserved for testing. The training set represents about 5 min of speech or 3000 phonemes. The test set contains 270 phonemes. Both the test and the training set are hand labeled and segmented, using a 61-symbol phonetic alphabet to represent the different acoustic realizations of 42 phonemes in English. The speech is sampled at 20 kHz, and frames of speech are analyzed every 10 ms using a 20 ms Hamming window. A mel-warped cepstral analysis is used, resulting in a vector of the first 14 mel-warped cepstral coefficients for each analysis interval. The zeroth cepstral coefficient, which primarily represents energy, is not used.

Hand-Segmented Training and Recognition Results

In order to more efficiently evaluate different parameters of the segment model, the initial segment-based phoneme recognition results are based on hand-segmented input speech for both training and test. In this section, we discuss experiments involving variations in the pattern

match criterion (distance versus probability), the resampling algorithm, the phoneme covariance structure, and the use of a duration feature. There are no insertion rates reported in this section because the segmentations are known, so there are no insertions or deletions of phonetic events in the recognized phoneme string.

The first results compare phone-dependent diagonal and block-diagonal covariance structures as a preliminary measure of the usefulness of the block-diagonal covariance structure. Also, for comparison, we include the performance of the Euclidean and Mahalanobis minimum distance pattern match algorithms (5). Table I summarizes the results for the log Gaussian probability and Mahalanobis distance. The Euclidean distance is equivalent to a Mahalanobis distance with a covariance matrix equal to the identity matrix. All results are based on linear time-sampling with interpolation. The three results demonstrated that: 1) using more detailed covariance structure improves recognition, even for relatively little training data (from 1 to 120 observations per model), and 2) the Gaussian pattern match is slightly better than the Mahalanobis distance. The only difference between the Mahalanobis distance and the Gaussian model is the use of the determinant of the covariance in the pattern match score. All remaining experiments will assume a Gaussian pattern match and a phone-dependent block-diagonal covariance.

Next, we looked at different segment resampling techniques. Using interpolation, we compared linear time sampling and space sampling, using the weighted probability in the case of space sampling. The weighted probability (6) improves the performance somewhat (2 percent) for space sampling. Space sampling with interpolation produces slightly better results than does linear time sampling with interpolation, as predicted by the segment quantizer results [4]. Next, we compared the two warping techniques without using interpolation. Here, the linear time-sampling result improves, while the space-sampling result does not. The most promising result is gained using linear time-sampling without interpolation. Consequently, further experiments are based on linear time sampling without interpolation, unless otherwise specified.

The improvement in performance associated with not interpolating samples is probably because interpolation introduces some undesirable spectral smearing. For example, interpolating over a region of large spectral variation, such as between a burst and a vowel onset, might result in nonspeech-like spectra. That interpolation introduces spectral smearing can be shown by plotting the template means and variances for both cases. In most cases, the means and variances are close for the different phonemes, but there are several phonemes which show a clear increase in variance for at least the first few cepstral coefficients. These phonemes are "IY," "IH," "UW," "F," "S", and "TH."

We also considered a few different covariance estimates for the Gaussian models as a study in robust algorithms for covariance estimates. Different combinations of using

TABLE I
RECOGNITION PERFORMANCE FOR THE MAHALANOBIS DISTANCE AND THE LOG GAUSSIAN PROBABILITY AS A FUNCTION OF COVARIANCE STRUCTURE

Covariance Structure	% Phonetic Recognition	
	Mahalanobis Dist	Gaussian Prob
Euclidean (W=I)	62.6	—
Diagonal	66.7	70.0
Block Diagonal	70.7	73.3

phone-dependent or phone-independent covariances, block-diagonal or diagonal covariances, were evaluated. All results were based on linear time sampling without interpolation. In the subset of phonemes infrequently observed (less than 30 times), we were able to improve recognition performance from 14 to 78 percent by using different combinations of covariance estimates. Looking at the performance for the entire phoneme set, recognition performance improved from 68.1 percent for using only phone-independent covariances to 78.5 percent for the best case covariance estimates. This best result will be referred to as the robust covariance estimate and is based on the algorithm described in Section IV, which uses progressively less complex covariance structures as the number of observations for a phoneme decreases. The phone-independent covariance is considered the simplest structure. The recognition results for the best case (robust) phone-dependent result and the worst case phone-independent result are given in Table II.

Finally, we investigated the use of segmental features by including a duration feature in each sample. Duration feature experiments involved comparing several different variations of the sample duration feature for both linear time sampling and space sampling. All experiments used the robust phone-dependent covariance structure. Among the different variations, we compared percent duration and duration in numbers of frames, space sampling and time sampling, and sampling with and without interpolation. Of the different combinations, the best results were achieved using sample duration in frames w_j (7) and linear time sampling without interpolation. Using presegmented inputs to the recognition algorithm, performance improved from 78.5 to 80.4 percent as a result of using the duration feature (Table II). As an aside, the motivation for using duration as a feature was to distinguish long vowels from short vowels ("IY" and "IH") and voiced from unvoiced fricatives ("V" from "F"), for example. This did not turn out to be the case. In fact, the main improvement in performance due to using a duration feature was a reduction in the error rate for the phone "schwa."

Results for Automatic Algorithms

In the above results, we assumed knowledge of phoneme segmentation for both training and recognition. In a practical system, we would not have high-accuracy phoneme segmentations for recognition, and we do not want to be restricted to using hand-segmented speech for train-

TABLE II
RECOGNITION RESULTS FOR SOME VARIATIONS OF THE BASIC SEGMENT MODEL USING MANUALLY SEGMENTED DATA

Model Variation	% Recognition
Phone-Independent	68.1
Phone-Dependent	78.5
Phone-Dep + Duration	80.4

ing. We start by presenting experimental results for joint recognition and segmentation, which does not require segmented input speech. The models for these results, however, are trained from hand-segmented speech. Next, we give results for the fully automatic system.

Joint Segmentation and Recognition: The results for joint segmentation and recognition of speech are given in Table III for two cases: the basic model using linear time sampling without interpolation, and the extension which uses the duration feature. The cost per segment (13) was chosen so that the insertion rate would be approximately 10 percent. From these results, we conclude that we can expect a 4–5 percent loss in recognition performance, accompanied by a 10 percent insertion rate in moving from a presegmented input to joint segmentation and recognition.

Although dynamic programming is an efficient technique for joint optimization of segmentation and phoneme recognition, it is still computationally expensive. To reduce the computational cost somewhat, we only allowed phoneme segments to end on every other frame (that is, $s(i)$ must be even), reducing the computation by a factor of four. This reduction of time resolution seemed to hurt the recognition performance slightly. For example, there is about 2–3 percent degradation in performance when moving from a time resolution of two to a time resolution of three. No experiments were run using a time resolution of one.

Automatic Training: Results for the iterative training algorithm are summarized in Table IV. The table gives the automatic recognition performance on the test data after each iteration of the algorithm, which can be compared to the automatic recognition performance using hand-segmented speech in training. The experiments used linear time sampling without interpolation and robust covariance estimates.

As in the automatic recognition experiments, computation for automatic segmentation can be reduced by a factor of four by using a time resolution of two frames. For automatic segmentation, we ran some experiments on a subset of the training set to determine the difference in segmentation error due to using time resolutions of one, two, and three. Again, the difference in average segmentation error associated with the different time resolutions is small, so we chose to use a time resolution of two in all experiments. Note that the best possible average segmentation error using a time resolution of two frames is 0.5.

The initial estimate for the phonetic models was designed from ten sentences of hand-segmented speech. The

TABLE III
RECOGNITION RATES USING MANUALLY SEGMENTED SPEECH COMPARED TO RECOGNITION/INSERTION RATES FOR JOINT SEGMENTATION AND RECOGNITION FOR TWO CASES: THE BEST CASE PHONE-DEPENDENT MODEL TRAINED WITH MANUALLY LABELED SPEECH AND THE SAME MODEL WITH A DURATION FEATURE

Model Variation	% Recognition/% Insertion	
	Hand-segmented	Auto-segmented
Phone-Dependent	78.5/0.0	74.4/10.0
Phone-Dep + Duration	80.4/0.0	75.9/10.7

TABLE IV
AUTOMATIC RECOGNITION RESULTS (USING JOINT RECOGNITION AND SEGMENTATION) AND AUTOMATIC SEGMENTATION ERROR FOR A FEW ITERATIONS OF AUTOMATIC TRAINING, COMPARED TO TRAINING BASED ON HAND-SEGMENTED SPEECH

Iteration	% Recognition	% Insertion	Seg. Error
Hand-segmented	74.4	10.0	0
Initial estimate	61.1	5.9	1.55
First pass	73.0	7.8	1.48
Second pass	73.0	9.3	1.55
Third pass	73.7	7.8	—

model for the phonemes that were never observed in these sentences was defined to be a general (phone-independent), sample-dependent segment model, estimated using all resampled-segment training observations. The initial set of phonetic models was then used to segment all of the remaining 90 training sentences. The new segmentations were then used to design new models, and so forth. The recognition results on the test sentences and the average segmentation error (relative to hand segmentations) of the training data for each pass are given in Table IV. The results show that the automatic training algorithm very quickly (after only one pass) yields models which are close in performance to the models based on hand-segmented data. After three training passes, the performance of the automatically trained models is equivalent to the performance of models trained on hand-segmented data. The average segmentation error of the training speech relative to manually marked segmentations was about 1.5 frames/segment. Fig. 3 illustrates the convergence of the mean of the m samples of the first cepstral coefficient for three phonemes, two of which were not observed in the initial training set.

For reference, the discrete hidden Markov model performance on these data for a phoneme system is 62 percent with 12 percent insertions [2]. HMM recognition performance on this database is higher when using context-dependent models. By context-dependent models, we mean there are several different models for each phoneme, each depending on the specific phonetic context. For example, the phoneme "AH" would have separate models for the subclass that occurred in the context of "R" and "T." Using models conditioned on the class of the neighborhood phonemes, there would be a total of 61^2 different left-context models or 61^3 different left-right-

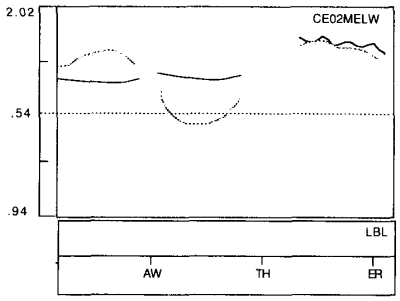


Fig. 3. Trajectory of the mean first cepstral coefficient for three phonemes after the initial guess (solid line), one pass of training (dashed line), and three passes of training (dotted line). The first two phonemes were not observed in the initial training set, and thus have the general phoneme mean after the initial guess.

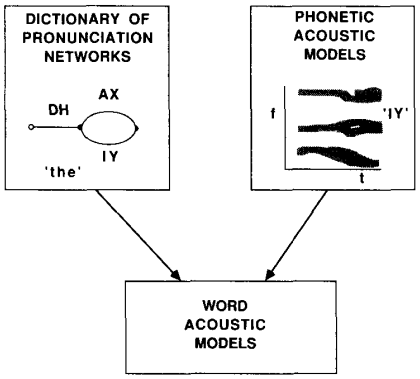


Fig. 4. Acoustic word model derived from a phonetic pronunciation network and phoneme acoustic models.

context models. Only those contexts seen in training will be modeled, so the actual number of models used will be smaller than the total possible number of models. Using phonetic models conditioned on the left context of the phoneme, the phoneme recognition performance on these data is 75 percent correct with 12 percent insertions [2]. Both HMM results are based on a system using three-state discrete HMM phone models and 256 spectral templates for input frame quantization. It is interesting to note that the segment phoneme system with 61 phoneme models performs about as well as the HMM system with 600 phoneme and left-context models. This result does not prove that the segment model is more useful than HMM's since the segment model is a ten-state continuous distribution phoneme model while the HMM is a three-state discrete distribution phoneme model. Nevertheless, it does demonstrate that the segment model is a potentially useful technique for speech recognition since HMM's are possibly the most successful technique for speech recognition currently available.

To evaluate the significance of the phoneme recognition results, we determined confidence of the error rate using a binomial model. For a test size of 270 phonemes, the 90 percent confidence interval for a phoneme error rate of 26 percent is ± 4.4 percent. In addition, there is greater than 99 percent confidence that the segment model outperforms the HMM phone model.

VI. Word Recognition

In this section, we will describe the segment-based word recognition system and present results for a speaker-dependent, continuous speech, 350-word vocabulary task.

The segment-based word recognition system consists of a dictionary of phonetic pronunciation networks and a collection of segment phonetic models. A word model is built by concatenating phonetic models according to the pronunciation network (Fig. 4). The recognition algorithm is simply a dynamic programming search (Viterbi decoding) of all possible word sequences for the best scoring word sequence. For the results in this paper, we assume that words are independent and equally probable; there is no grammar (statistical or deterministic) associated with recognition. Within each word, we use dynamic programming to find the best phoneme segmentation for that word, where the phoneme sequence is constrained by the word pronunciation network. The training algorithm for the phoneme models used in the word recognition system is no different from the phoneme system training algorithm. Implementing a segment-based word recognition system is equivalent to implementing the phoneme recognition algorithm with a finite-state grammar constraint on the sequence of phonemes based on the word pronunciation network.

The segment models used in the word recognition experiments used linear time warping without interpolation, independent samples, and cepstral features only (no duration feature).

For word recognition, we used a 350-word vocabulary, speaker-dependent database based on an electronic mail task. We present results for three different male speakers. 15 min of speech was used for training the 61 phonetic models for each speaker, from which the word models were built. An additional 30 sentences for each speaker (187 words) are used for testing recognition accuracy. Given three speakers, the total test set then has 561 words. Analysis parameters are the same as for the previous database. Since we do not use a grammar, homophones (such as "two" and "to") are indistinguishable, and word error rates reported do not include homophone confusion as errors.

The initial segment model is based on training from segmentations given by a discrete HMM recognition system which has an estimated average segmentation error of approximately two frames per segment. After one pass of training, using a time resolution of two frames for automatic segmentation, the average recognition rate increased from 77 percent correct with 3.9 percent insertions to 83 percent correct with 3.7 percent insertions. The recognition/insertion results after one pass of training for the three speakers are summarized in Table V with the HMM recognition results for comparison. Again, the segment results are based on linear time sampling with no interpolation. The HMM results are based on five passes

TABLE V
WORD RECOGNITION/INSERTION RESULTS FOR THREE SPEAKERS UNDER THREE DIFFERENT SYSTEMS: THE PHONEME-BASED SEGMENT SYSTEM, THE PHONEME-BASED HMM SYSTEM, AND THE CONTEXT-BASED (PHONEME, LEFT, AND RIGHT CONTEXT) HMM SYSTEM

Speaker	Segment-PH	HMM-PH	HMM-PH-LE-RI
RS	87/5.3	85/10.2	90/1.1
FK	83/2.1	75/ 5.4	88/2.7
AW	78/3.7	68/ 7.5	86/3.7
Average	83/3.7	76/ 7.7	88/2.5

of training with the forward–backward algorithm. The segment phoneme system outperforms the phoneme-based HMM system, but the segment phoneme system does not quite match the HMM context model system. This suggests that context-dependent segment models might be useful. Note that in the earlier phoneme results, the segment system matched the performance of HMM models conditioned on left context only. Here we give results for HMM models conditioned on both left and right context. The HMM system with context models conditioned on both left and right context uses 2000 models or 35 times the number used by the segment system. Equivalently, the segment-based system cuts the error rate (counting insertions as errors) by one-third compared to the HMM system with the same number of models. Again, we do not claim that these results show that the segment model is superior to HMM's. We merely use the HMM results as a reference point to make the segment recognition results more meaningful.

To evaluate the significance of the word recognition results, we determined the confidence of the error rate using a binomial model. With three speakers, and therefore a total of 561 words, the 90 percent confidence interval for the average segment error rate of 20 percent is ± 2.8 percent. When we compare the different systems, we find that the confidence that the segment model is better than the HMM phone system is greater than 99 percent, and the confidence that the HMM context system is better than the segment phone system is also greater than 99 percent.

VII. COMPLEXITY

The stochastic segment model is a more detailed phoneme model and is, therefore, more complex. To put the complexity in perspective, it is a useful exercise to estimate the computational and storage requirements of the segment recognition algorithm. For reference, these estimates are compared to the requirements of a discrete HMM recognition system.

We begin by summarizing the system configuration and compromises made for both training and complexity reasons. The resampling technique is linear time warping without interpolation, which has negligible complexity. The model is based on the assumption of sample independence (10 14 $\times$ 14 covariance matrices) because of training limitations, but this also reduces both the computation and storage requirements significantly. The duration feature was not used because it increases computational complexity as described below.

First we will describe the computational requirements of a segment-based recognition system. In the dynamic programming approach to joint segmentation and recognition, at each point in time, we search for the best phoneme ending at this time. If we restrict phonemes to a maximum duration of 500 ms and constrain segmentations to a time resolution of t, there are $50 \times 61 \times 100/t^2$ segment probability computations per second in phoneme recognition, each of which corresponds to m k-dimensional weighted distance computations. This computation can be reduced by a factor of two with no loss in performance by pruning out low-scoring theories. Using a table lookup for precomputed sample distances reduces computation by approximately a factor of seven. Note that the table lookup only reduces computational complexity if segmental features (such as duration) are not used. In word recognition, we also use a table to store previously computed phoneme scores in order to avoid computing phoneme scores multiple times because they appear in multiple words. With the computation reduction, the recognition takes about 200 times real time on a time-shared VAX 11-780 (a 0.25 MFLOPS LINPACK single-precision machine). Word recognition on a Symbolics 3670 Lisp machine takes approximately 400 times real time, using the same computation reduction techniques as in phoneme recognition, except for pruning. For reference, an HMM decoding system on a Lisp machine, which also uses some pruning techniques for efficient computation, takes approximately 50 times real time for decoding an utterance. Since training is a one-time expense, we will not go into detail on the computational requirements for training. We will simply mention that automatic segmentation of an utterance requires approximately half the computation of joint segmentation and recognition, and one iteration of segment training is roughly equivalent to five iterations of HMM training using the forward–backward algorithm.

The storage requirement for the segment phoneme models is on the order of Nmk^2 real numbers where N is the number of models, m is the number of samples/segment, and k is the dimension of each sample. (Note that these values reflect the storage requirement of a block-diagonal covariance structure. The storage requirement of the full covariance would be significantly higher, approximately $N(mk)^2$.) The HMM storage per model for the three-state discrete HMM in the BBN Byblos system [2] is roughly equivalent to the segment model storage per model, but the HMM system requires an order of magnitude more models to achieve the same performance as the segment model.

VIII. CONCLUSIONS

To summarize, this paper introduces a new approach to acoustic phonetic modeling which is based on jointly modeling all features of a speech segment corresponding to a phoneme. The model consists of 1) a transformation

which resamples the variable-length realization of a phoneme to a fixed-length segment, and 2) a multivariate Gaussian model for the resampled segment. Automatic recognition and training algorithms are given which result in system performance close to that of a system based on manually segmented speech. The results presented here demonstrate that the segment model offers the potential for large improvements in both phonetic and word recognition in continuous speech. A segment-based word recognition system reduces the word error rate by one-third in comparison to an HMM word recognition system with the same number of phonetic models. Similar performance results were demonstrated in a phoneme recognition system comparing the segment model to the HMM. The complexity of the segment recognition algorithm is greater than HMM recognition complexity, but of the same order of magnitude. Moreover, the probability computations are vector operations, so the algorithm is amenable to parallel algorithms.

Future research in segment modeling includes designing more detailed segment models, specifically, context-dependent models. A context-dependent segment model is a segment model for a phone conditioned on the left and/or right phonetic context. Since there are few observations (if any) of most context-dependent events, robust training becomes an important issue in context-dependent phonetic modeling. Initial results show that without improved training algorithms, context-dependent segment models do not offer significant improvement in performance over context-independent models.

Acknowledgment

The authors thank Y.-L. Chow, O. Kimball, F. Kubala, P. Price, and R. Schwartz for their help in implementing the segment word recognition system. Thanks also go to J. Makhoul and A. Wilgus for helpful comments in reviewing the paper.

References

[1] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, pp. 179–190, Mar. 1983.

[2] R. M. Schwartz, Y. L. Chow, O. A. Kimball, S. Roucos, M. Krasner, and J. Makhoul, "Context-dependent modeling for acoustic-phonetic recognition of continuous speech," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 1205–1208, Paper 31.3.

[3] Y. L. Chow, R. M. Schwartz, S. Roucos, O. A. Kimball, P. J. Price, G. F. Kubala, M. O. Dunham, M. A. Krasner, and J. Makhoul, "The role of word-dependent coarticulatory effects in a phoneme-based speech recognition system," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tokyo, Japan, Apr. 1986, pp. 1593–1596, Paper 30.9.

[4] S. Roucos, R. Schwartz, and J. Makhoul, "Segment quantization for very-low-rate speech coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Paris, France, May 1982, pp. 1565–1569.

[5] C. Tsao and R. M. Gray, "Matrix quantizer design from LPC speech using the generalized Lloyd algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 537–545, June 1986.

[6] E. Bocchieri and G. Doddington, "Frame-specific statistical features for speaker-independent speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 755–764, Aug. 1986.

[7] G. Kopec and M. Bush, "Network-based isolated digit recognition using vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 850–867, Aug. 1985.

[8] L. R. Bahl, R. Bakis, P. S. Cohen, A. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Continuous parameter acoustic processing for recognition of a natural speech corpus," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Atlanta, GA, Apr. 1981, pp. 1149–1152.

[9] S. Roucos, M. Ostendorf, H. Gish, and A. Derr, "Stochastic segment modeling using the estimate-maximize algorithm," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, Apr. 1988, pp. 127–130.

[10] B.-H. Juang and L. R. Rabiner, "Mixture autoregressive hidden Markov models for speech signals," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1404–1413, Dec. 1985.

[11] S. Makino and K. Kido, "Recognition of phonemes using time-spectrum pattern," *Speech Commun.*, vol. 5, pp. 225–238, June 1986.

[12] L. R. Rabiner, M. M. Sondhi, and S. E. Levinson, "A vector quantizer incorporating both LPC shape and energy," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, Mar. 1984, Paper 17.1.

[13] H. Murveit and M. Weintraub, "1000-word speaker-independent continuous-speech recognition using hidden Markov models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, Apr. 1988, pp. 115–118.

[14] K.-F. Lee and H.-W. Hon, "Large-vocabulary speaker-independent continuous speech recognition using HMM," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, New York, NY, Apr. 1988.

[15] M. A. Bush and G. E. Kopec, "Network-based connected digit recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-35, pp. 1401–1413, Oct. 1987.

[16] D. K. Burton, J. E. Shore, and J. T. Buck, "Isolated word recognition using multisection vector quantization codebooks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 837–849, Aug. 1985.

[17] C. W. Helstrom, *Statistical Theory of Signal Detection*. Oxford, England: Pergamon, 1975.

Mari Ostendorf received the B.S., M.S., and Ph.D. degrees in 1980, 1981, and 1985, respectively, all in electrical engineering from Stanford University, Stanford, CA.

In 1985 she joined the Speech Signal Processing Group, BBN Laboratories, where she worked on low-rate coding and acoustic modeling for continuous speech recognition. In 1987 she joined the Faculty at Boston University, Boston, MA, where she is currently an Assistant Professor in the Department of Electrical, Computer, and Systems Engineering. Her research interests include data compression and recognition, particularly in speech processing applications.

Dr. Ostendorf is a member of Sigma Xi.

Salim Roukos was born in Batroun, Lebanon, on May 5, 1954. He received the B.E. degree from the American University of Beirut, Lebanon, in 1976, and the M.Sc. and Ph.D. degrees from the University of Florida, Gainesville, in 1977 and 1980, respectively, all in electrical engineering.

In 1989 he joined the Computer Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, NY, to work on statistical language modeling. He was also an Adjunct Professor at Boston University in 1989. From 1980 to 1989 he was with BBN Laboratories, Cambridge, MA, where he developed algorithms for speech compression at 300 bits/s, speaker identification, continuous speech recognition, and speech understanding. He has also filed for a patent on time-scale modification. His research interests include statistical modeling and pattern recognition.

Dr. Roukos is currently Chair of the Digital Signal Processing Technical Committee of the IEEE Acoustics, Speech, and Signal Processing Society.