A Method for the Construction of Acoustic Markov Models for Words

L. R. Bahl, *Member, IEEE,* P. F. Brown, P. V. de Souza, R. L. Mercer, *Member, IEEE,* and M. A. Picheny, *Member, IEEE*

Abstract— **A new technique for constructing Markov models for the acoustic representation of words is described. Word models are constructed from models of sub-word units called fenones. Fenones represent very short speech events, and are obtained automatically through the use of a vector quantizer. The fenonic baseform for a word—i.e., the sequence of fenones used to represent the word—is derived automatically from one or more utterances of that word. Since the word models are all composed from a small inventory of sub-word models, training for large-vocabulary speech recognition systems can be accomplished with a small training script. A method for combining phonetic and fenonic models is presented. Results of experiments with speaker-dependent and speaker-independent models on several isolated-word recognition tasks are reported. Comparative results with phonetics-based Markov models and template-based DP matching are also given.**

I. Introduction

THE POPULARITY of Markov models for acoustic modeling in speech recognition systems has grown considerably during the last two decades. In this paper we present a new method of constructing acoustic Markov models. We describe a method for deriving an acoustic representation of a word automatically from sample utterances of the word. This method results in a substantial decrease in recognition error rate when compared with methods based on phonetic representations of words. This method has been used in the Tangora large-vocabulary natural-language isolated-word speech recognition systems developed at the IBM Research Center [1], [2].

First, let us consider word-based Markov models, which were first suggested by Vintsyuk [3]. An example of a recognition system that uses Vintsyuk-type word-based Markov models is the system implemented by Bakis [4], [5]. Here, each word is represented by a Markov model that is derived from sample utterances of the word. The number of states in the model for a word is equal to the average duration of the word in frames. The frame size in Bakis's system is 10 milliseconds, and the average number of states for a word is about 30. Fig. 1 shows an example of a Vintsyuk-type model. The direct path through the model represents the average duration of the word. The self-loops allow for elongation of the word, and the transitions that skip around states allow for shortening of the word. There is a probability associated with each transition, and an output distribution associated with each state. The transition probabilities and output distributions for each word are estimated from several sample utterances of the word, using the forward-backward parameter estimation algorithm [5]–[7]. This system worked very successfully on a 250-word "Raleigh language" [5] continuous-speech speaker-dependent recognition task. However, there is one major drawback to such a system: the user must provide several sample utterances of each word to train the system, so it is not practical for large-vocabulary systems. The number of Markov parameters and the training data requirements grow linearly with the vocabulary size.

More recently, Rabiner and Levinson [8] described a system in whichthe number of states in each word model was reduced to about 5. This results in a substantial reduction in the number of parameters, without much degradation in the accuracy of the model. This is because neighboring states in the Vintsyuk model tend to be quite similar, and reducing several similar consecutive states into a single state does not degrade the model very much. However, training each word model requires several sample utterances, so the training data size still grows linearly with vocabulary size.

Another way to reduce the number of parameters is to build word models from a small inventory of sub-word models, such as phones, diphones, syllables, demi-syllables, etc. Most attempts at obtaining sub-word units are based on linguistic or phonetic concepts. An example of a system in which phones are used as the building blocks for constructing word models is described by Bahl, Jelinek, and Mercer [9]–[11]. Each word is represented by a sequence of phones, called the phonetic baseform of the word. A Markov model is established for each phone. The Markov model for a word is obtained by replacing each phone in the baseform for the word by its Markov model. Fig. 2 shows an example of a Markov model for a phone, a phonetic baseform, and the resulting Markov model for the word from a phonetics-based recognition system at IBM. The transitions drawn in dotted lines result in no output. In this example, the structure of the Markov models for all the phones is identical; the transition and output probabilities will of course be different for each phone. Different models may be used for each phone if desired. If a word has multiple pronunciations, then phonological rules can be used to expand the baseform into a graph representing the various pronunciations of the word [12]. Each phone can then be replaced by its Markov model to produce a Markov model of the word. This decomposition of words into phone sequences removes the dependence of the training data size on the vocabulary. In order to train the system adequately, one

Manuscript received October 29, 1987; revised October 23, 1991. The associate editor coordinating the review of this paper and approving it for publication was Dr. B. H. Juang.

The authors are with the Speech Recognition Group, IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598.

IEEE Log Number 9210873.

1063–6676/93$03.00 © 1993 IEEE

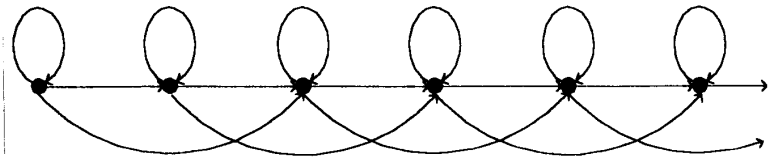

Fig. 1. Word-based Markov models used by Vintsyuk and Bakis.

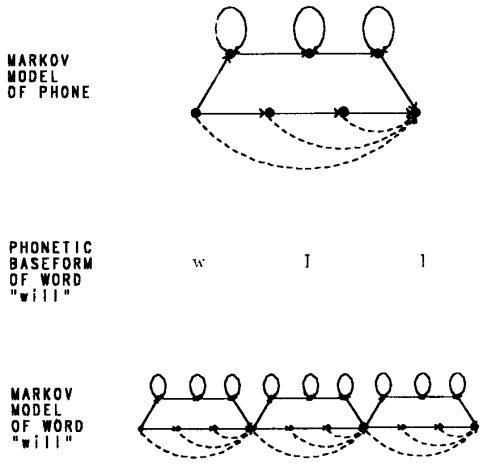

Fig. 2. Example of phone model, phonetic baseform, and word model.

now only needs several samples of each phone, and not several samples of each word. Phonetics-based models, however, have the drawback that the phonetic baseform is based on "expert" human knowledge, which unfortunately is subjective, difficult to extract, and subject to error.

When sufficient data is available to train them, the Vintsyuk-type word-based models used by Bakis outperform the phonetics-based models [13]. The performance of phonetics-based models can be improved by using context-dependent phone models [14]–[16]. However, on isolated word speech it is doubtful that they will perform as well as word-based models. The main reason for the superiority of the word-based models is that they model the words at a much finer level of detail. The effects of context-dependence and coarticulation within a word are implicitly built into the models. These observations do not necessarily carry over to continuous speech systems, where it is important to model the coarticulatory effects across word boundaries.

Our aim is to construct Markov models for isolated word speech that retain the accuracy of the word-based models while requiring no more training data than the phonetics-based models.

We can think of phones as a partition of the acoustic space into regions. The partition is based on *a priori* human linguistic knowledge. A phonetic baseform of a word is, then, a specification of the sequence of acoustic regions traversed in pronouncing that word. Instead of using phones, we use a partition of the acoustic space that is derived automatically through the use of a vector quantizer. Based on this partition, we construct a new type of baseform called a fenonic baseform. These baseforms can be used to construct Markov word models in the same way that phonetic baseforms are used for constructing word models. The performance of fenonic baseforms was tested on several isolated-word tasks. In Sections II and III, we describe some experiments with speaker-dependent fenonic baseforms, derived from single or multiple utterances of a word. Speaker-dependent baseforms, of course, do not alleviate the need for large amounts of training data from ech speaker. This problem is addressed in Section IV, where we discuss speaker-independent fenonic baseforms. In Section V, we summarize our work. In Section VI, we give a short summary of some related work by other researchers.

II. Singleton Speaker-Dependent Fenonic Baseforms

We will assume that readers are familiar with the basic concepts of vector quantization as used in speech recognition systems. Articles by Gray [17] and Makhoul, Roucos, and Gish [18] contain excellent surveys of vector quantization techniques. A vector quantizer (VQ) has as its input an acoustic waveform, and produces as its output a string of discrete labels. The input acoustic waveform is digitized, and a vector of acoustic parameters is extracted from the signal at regular intervals. This vector is then compared to a set of prototype vectors, and an acoustic label is produced that identifies the closed prototype vector. In our system, we extract a vector of 20 ear-model parameters [1], [19] from the speech signal at regular intervals of 10 milliseconds, and compare this vector to 200 VQ prototype vectors using Euclidean distance.

The 200 VQ prototypes are obtained in a completely unsupervised manner using the clustering algorithm of Linde, Buzo, and Gray [17], on 5 minutes of speech provided by a talker. Each VQ prototype has a label name associated with it. These label names can be arbitrary; e.g., the integers 1 to 200 would a reasonable choice. However, it is useful to assign label names that can be interpreted by humans. To do this, we used the VQ prototypes to label some speech data that had previously been phonetically labeled. Each VQ prototype was then assigned a label name that identified the phonetic phone that occurred most often in conjunction with the VQ prototype. Since several different VQ prototypes may get the same phonetic label, a number was appended to the phone name to make all the VQ label names distinct. The names are of no consequence to the recognition system; they are merely an aid to humans who may wish to examine data labeled with the VQ prototypes.

Let $F = \{f_1, f_2, \cdots, f_{N_F}\}$ be the alphabet of labels produced by the acoustic processor. Let F^* represent the set of all finite length strings constructed by concatenating

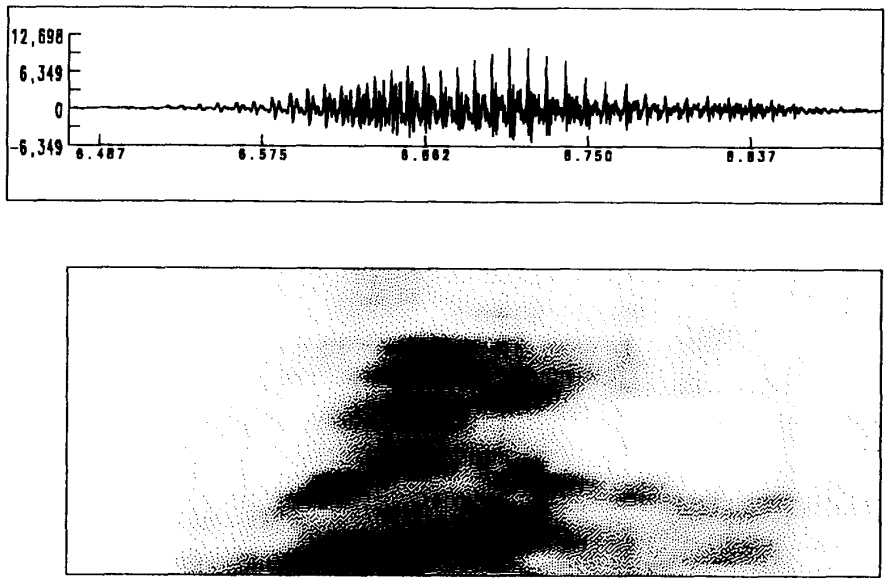

Fig. 3. Time waveform, spectral representation, and label string for the word *will*.

elements of F. Let $W = \{w_1, w_2, \cdots, w_{N_W}\}$ be the words in the vocabulary of the system. Let $w \in W$ be some word in the vocabulary, and let $y_{1 \to l} = y_1, y_2, \cdots, y_l \in F^*$ be the label string produced by the acoustic processor in response to a speaker uttering the word w. Fig. 3 shows the time waveform, a spectral representation, and the label string for one utterance of the word *will* pronounced in isolation. The SIL label corresponds to silence. The label string corresponding to the word *will* in Fig. 3 is W1 W2 W1 W2 W2 W3 W4 OU1 OU1 OU2 UH1 UH1 I1 I1 I1 I2 I1 I1 UH2 UH1 OU3 L1 L1 L2 L2 L2 L2 L2 L2 L2 W3 L2 W3 W2 W2.

Roughly speaking, the first 10 labels belong to the phone /w/, the next 10 to the phone /I/ and the last 15 to the phone /l/. Note that 6 different labels occur in the label string for /w/, 4 for /I/, and 5 for /l/. Note also that the labels W2 and W3 occur in both /w/ and /l/.

We can treat the label string $y_{1 \to l}$ as if it were a baseform for the word w. Of course, other utterances of the same word will not result in label strings that are identical to $y_{1 \to l}$, but will generally produce label strings that are similar to $y_{1 \to l}$. We can model this variation by replacing each label in the baseform by a Markov model. What we have done here is to replace the phonetic alphabet used in making phonetic baseforms by an alphabet of label-based sub-phone units (called fenones) which is used to make a new type of baseform called a fenonic baseform. Let $P = \{p_1, p_2, \cdots, p_{N_P}\}$ represent the alphabet of fenones. There is an obvious one-to-one correspondence between the elements of fenone alphabet and the underlying label alphabet. Since each fenone represents a very short acoustic interval, the variation can be adequately modeled by very simple Markov models of the type shown in Fig. 4. A 2- or 3-state model with 3 or 4 transitions suffices to account for the variation in a fenone, while a larger model, such as the 7-state 13-transition model shown in Fig. 2, is needed to adequately model the variation in a phone. The Markov model for a word is derived by concatenating the Markov models of the fenones in its fenonic baseform. In Fig. 4(c), the lowest transition for fenone n connects to the initial state of fenone $n + 2$, so concatenating these models results in a Vintsyuk-type word model. The simplest model, shown in Fig. 4(a), allows for deletion and unlimited elongation of a fenone. The model in Fig. 4(b) limits a fenone from producing more than 2 labels. The model in Fig. 4(c) allows a maximum shortening by a factor of 2. Obviously, many other models are possible. All the experimental results in this paper were obtained using a fenone model of the type in Fig. 4(a). Experiments were also carried out with models of the type shown in Figs. 4(b) and 4(c) with no significant differences in recognition accuracy.

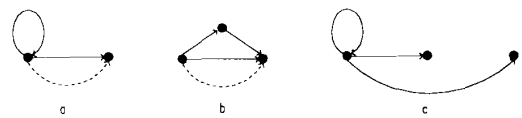

Fig. 4. Examples of Markov models for fenones.

By taking one instance of each word in the vocabulary, we canproduce fenonic baseforms for all words in the vocabulary. A Markov model for a word is constructed by concatenating the elementary Markov models of the fenones in its fenonic baseform. Note that the number of parameters depend only on the size of the fenone alphabet, and not on the size of the vocabulary. Baseforms constructed in this manner are called singleton fenonic baseforms, because they are constructed from one sample utterance of each word.

Fenonic baseforms are much longer than phonetic baseforms. In our experiments they were, on the average, about 10 times as long as phonetic baseforms. However, since the Markov models for fenones are substantially simpler than Markov models for phones, the fenonic and phonetic word models have roughly the same number of states.

Before the fenonic models can be used for recognition, it is, of course, necessary to train the transition and output probabilities of the fenones. This is accomplished by using additional training data from the speaker. The training is done in the usual manner by the use of the forward-backward algorithm. The training data need not contain several samples of each word. Since all the words are composed of the same 200 fenones, there is an implicit "tying" amongst subparts of different words. It is this tying that makes it possible to train all the word models even if the training data does not contain samples of all the words in the vocabulary.

Fenonic baseforms model the acoustic trajectory of a word at a much finer level of detail than phonetic baseforms. In this respect, fenonic baseforms are similar to word templates used in template-based DP matching [20], [21]. However, there are some very important differences between fenonic baseforms and DP word templates. Each "frame" in a word template is represented by a vector of parameter values, whereas in a fenonic baseform it is represented by a single fenone. The lattice generated by a word model using the fenone model of Fig. 4(a) is the same as the lattice used in conventional template-based DP matching. But, whereas the transition and output probabilities associated with the Markov models of fenones are trained to the speaker, no such training process exists in template-based DP matching.

Since a singleton baseform is derived from a single sample of a word, it is questionable whether it is a robust representation of a word and will result in good recognition accuracy in a speech recognition system. To investigate this question we tested the performance of singleton fenonic baseforms on two isolated-word recognition tasks. For comparison, parallel recognition experiments were done with two other techniques: 1) phonetics-based models and 2) template-based DP matching. The DP matching experiments were done by Haltsonen [22], [23].

The phonetic and fenonic Markov models have roughly the same number of parameters to be trained. The phone alphabet was of size 55, and each phone was modeled by a 7-state 13-transition model of the type illustrated in Fig. 2. Only 3 distinct output distributions are used in each phone. The 3 leftmost output transitions are tied and model the beginning of the phone; the center 4 are tied and model the central part of the phone; and the rightmost 3 are tied and model the end of the phone. Thus there are $165(55 \times 3)$ output distributions and 715 (55×13) transition probabilities to be estimated in the phonetic Markov models. The fenone alphabet was of size 200, and each fenone was modeled by a 2-state 3-transition model of the type illustrated in Fig. 4(a). The output distributions on the two output producing transitions were tied. Thus there are 200 output distributions and $600(200 \times 3)$ transition probabilities to be estimated in the fenonic Markov models.

All the experiments reported in this section were done with one male speaker. The speech was recorded in a quiet office environment using a close-talking Shure SM-12 microphone. The results of the experiments are shown in Table I.

The first task was the keyboard character recognition task havinga vocabulary of 62 words. The vocabulary includes the letters A–Z, the numbers 0–9, puncutation characters such as . and , and : and ; and other characters such as @ and # and % and & and $. The training data consisted of 10 utterances of each word. The first utterance of each word was used for creating a singleton fenonic baseform, and the remaining data was used for training the Markov models. In the DP matching experiment, the first instance of each word in the training data was used for constructing a template. The test data was 10 new utterances of each word. As can be seen in Table I, singleton fenonic baseforms had a much higher accuracy than phonetic baseforms. They performed slightly better than template-based DP matching, but this comparison is not entirely fair. In the fenonic baseform experiment all 10 training utterances were used (1 for baseform construction and the rest for Markov model training), whereas in the DP matching experiment only 1 training utterance was used.

TABLE I
PERFORMANCE OF SINGLETON FENONIC BASEFORMS, PHONETIC BASEFORMS AND DP MATCHING

Recognition Task	Acoustic Model	Language Model	Word Error Rate
keyboard	singleton fenonic	uniform	2.3%
keyboard	phonetic	uniform	4.7%
keyboard	DP matching	uniform	3.3%
OC2000 random words	singleton fenonic	uniform	7.4%
OC2000 random words	phonetic	uniform	5.2%
OC2000 random words	DP matching	uniform	7.7%
OC2000 natural sentences	singleton fenonic	word 3-gram	2.7%
OC2000 natural sentences	phonetic	word 3-gram	2.5%

The second task, known as OC2000, has a vocabulary consisting of the 2000 most frequent words in a database of IBM office correspondence. The training data consisted of 2 utterances of each word; the first was used for creating singleton fenonic baseforms and DP word templates. Homophonous words such as *to, too,* and *two* had identical baseforms and templates. The rest of the training data was used for training the Markov models. Two sets of test data were used. The first consisted of 1 utterance of each word in the vocabulary, spoken in a random order. The second test was on 100 natural sentences obtained from memos, comprising a total of 1297 words. On the random word sequences, a uniform language model that assigns equal probabilities to all words in the vocabulary was used. On the natural sentences, recognition was carried out with a 3-gram Markov language model. The language model was not incorporated into the template-based DP matching system, hence no results are reported for DP matching on the natural sentences. When no language model was used, the scores of homophonous words were identical, so there were no homophone errors. On this large-vocabulary task, we see that singleton fenonic baseforms are no longer superior to phonetic baseforms or template-based DP matching.

III. MULTI-SAMPLE SPEAKER-DEPENDENT FENONIC BASEFORMS

Since singleton fenonic baseforms offer no performance improvement on a large-vocabulary task, we investigated the possibility of using fenonic baseforms derived from multiple utterances of words. In order to do this, we need a method

for combining the information from several utterances into a single baseform. We start with singleton fenonic baseforms, and train the acoustic Markov models using some training data as in the previous section. As a result of this process we now have a set of fenones, P, complete with transition and output probabilities. We use these fenones as the units out of which the baseforms will be built. Thus each baseform will be a fenone sequence $p_{1\to m} \in P^*$, where P^* denotes the set of all finite-length sequences of elements of P. Let $y_{1\to l_1}^{(1)}, y_{1\to l_2}^{(2)}, \cdots, y_{1\to l_n}^{(n)}$ be the label strings corresponding to n different utterances of the word w. The optimal multi-sample fenonic baseform for w is defined to be the fenone sequence $\hat{p}_{1\to m} \in P^*$ which maximizes the probability of the n observed label strings for w, i.e.,

$$\hat{p} = \text{argmax}\, \Pr\{y^{(1)}, y^{(2)}, \cdots y^{(n)}|p\}. \quad (1)$$

For simplicity we have dropped the subscripts on y and p.

Assuming that the $y^{(i)}$ are conditionally independent of each other given p, we have

$$\Pr\{y^{(1)}, y^{(2)}, \cdots, y^{(n)}|p\} = \prod_{i=1}^{n} \Pr\{y^{(i)}|p\}. \quad (2)$$

From (1) and (2), we have

$$\hat{p} = \text{argmax} \prod_{i=1}^{n} \Pr\{y^{(i)}|p\}. \quad (3)$$

From (3) we can see that the problem of finding the optimal multi-sample fenonic baseform is very similar to the problem of maximum likelihood word sequence decoding as formulated in the papers by Bahl, Jelinek, and Mercer [9], [11], with the following two differences: 1) we search for the most likely fenone sequence rather than most likely word sequence, and 2) we have multiple label strings rather than a single label string. With minor modifications, the decoding algorithm of [9], [11], which carries out a tree search using a stack, can be used to search for the optimal fenonic baseform. The first modification is to replace the vocabulary of words by the vocabulary of fenones. The second modification is that the acoustic match is carried out for multiple label strings; the total acoustic match probability being the product of the individual acoustic match probabilities for the label strings. The Appendix contains a detailed description of the modified stack algorithm. Obviously, any other probabilistic decoding algorithm can be similarly adapted to search for the optimal fenonic baseform.

Recognition experiments on the keyboard and OC2000 isolated-word tasks were carried out using multi-sample fenonic baseforms. For comparison, template-based DP matching experiments were also carried out using a technique that uses multiple utterances for creating reference templates. One reference template was created for each word, using the averaging method described by Haltsonen [23].

Again, all experiments reported in this section were carried out with the same male speaker, and under the same conditions as described in the previous section. The results of these experiments are summarized in Table II. Column 3 indicates the number of sample utterances used in creating the fenonic baseforms and DP matching templates.

TABLE II
PERFORMANCE OF MULTI-SAMPLE FENONIC BASEFORMS, PHONETIC BASEFORMS AND DP MATCHING

Recognition Task	Acoustic Model	Number of Samples	Language Model	Word Error Rate
keyboard	multi-sample fenonic	10	uniform	0.8%
keyboard	phonetic		uniform	4.7%
keyboard	DP matching	10	uniform	0.8%
OC2000 random words	multi-sample fenonic	4	uniform	2.2%
OC2000 random words	phonetic		uniform	5.2%
OC2000 random words	DP matching	4	uniform	4.9%
OC2000 natural sentences	multi-sample fenonic	4	word 3-gram	0.7%
OC2000 natural sentences	phonetic		word 3-gram	2.5%
OC2000 natural sentences	multi-sample fenonic	4	uniform	8.6%
OC2000 natural sentences	phonetic		uniform	15.7%
OC2000 natural sentences	DP matching	4	uniform	18.0%

In the keyboard task, 10 utterances of each word were used to construct a multi-sample fenonic baseform and a DP reference template. When recognition was done with the multi-sample fenonic baseforms, the acoustic Markov model statistics were those obtained by training with the singleton fenonic baseforms. We also tried training the multi-sample fenonic baseforms directly with some additional training data, but the recognition rate did not change. From Table II, we can see that both multi-sample fenonic baseforms and template-based DP matching (0.8% error rate) performed much better than phonetic baseforms (4.7% word error rate).

In the OC2000 task, 4 utterances of each word were used forbaseform and template construction. In all the recognition experiments, the word error rate for multi-sample fenonic baseforms was substantially less than for phonetic baseforms or template-based DP matching.

IV. SPEAKER-INDEPENDENT MULTI-SAMPLE BASEFORMS

The baseforms used in Sections II and III do not alleviate the problem of reducing the amount of training data, since the speaker must provide one or several utterances of each word. Our goal was the construction of systems with large vocabularies—e.g., 5000 and 20 000 words—and it is out of the question that a user would be willing to provide even one sample of each word for vocabularies of this size.

In order to overcome this difficulty we investigated the possibility of using speaker-independent multi-sample fenonic baseforms. Experiments were carried out on the OC5000 isolated-word task, which is similar to the OC2000 task, except that the vocabulary is increased to include the 5000 most frequent words. Details of this task can be found in [1].

We collected 10 utterances of each word in the OC5000 vocabulary, one each from 10 different speakers. The speakers were hired from an agency that supplies temporary office help. All were male and natives of the New York–New Jersey area. Pooling together a few minutes of speech from each of the 10 speakers, a speaker-independent VQ label alphabet of size 200 was constructed. Then the entire speech database of 50 000 words was labeled with this alphabet. For each word, we chose at random one of the 10 utterances, and constructed singleton fenonic baseforms. Using the remaining 45 000 utterances, we trained the fenonic Markov models to establish the transition and output probabilities of the fenones to be used in the construction of the multi-sample baseforms. Using the label strings of all 10 utterances of a word, multi-sample fenonic baseforms were then constructed for each word in the OC5000 vocabulary. These baseforms were then used to recognize the speech of new talkers.

Once the baseforms had been constructed, the fenonic statistics used in the baseform construction process were discarded. A new user of the OC5000 word recognition system was required to read a training script of 100 natural sentences, comprising a total of about 1200 words. A new VQ label alphabet was established for each user, using 5 minutes of speech supplied by the user. Note that, unlike the case in the previous sections, the VQ label alphabet is no longer the same as the fenone alphabet. The 100 training sentences were labeled with the user's personal VQ prototypes. These label strings were then used to train the fenone Markov model statistics for the user. Recognition was performed on a test script of 50 sentences, comprising a total of 591 words. These sentences were selected at random from a collection of IBM internal office correspondence. Sentences containing words outside the vocabulary were rejected. Recognition was performed using a word 3-gram language model.

The recording conditions were the same as in the previous two sections, except that a Crown PZM desk-mounted microphone was used for recording all the speech. As stated earlier, 10 male talkers provided the data for constructing the baseforms. Table III summarizes the results of recognition experiments with 8 new talkers (4 male, 4 female) on the OC5000 task. The use of fenonic models reduces the average word error rate to 2.5% from the 3.5% error rate for phonetic models.

In some preliminary recognition experiments, we noticed that the phonetic and fenonic models tend to make different types of errors. For example, phonetic models have difficulty with words that contain strong coarticulatory effects. Fenonic models perform much better on these words since the coarticulation is implicitly better modeled in the fenonic baseforms. The fenonic models, on the other hand, make more word-ending errors—e.g., recognizing *concern* for *concerned,* and vice versa. The phonetic baseforms for these words

TABLE III
PERFORMANCE OF PHONETIC, SPEAKER-INDEPENDENT MULTI-SAMPLE FENONIC, AND COMBINED BASEFORMS ON OC5000 TASK

Speaker	Word Error Rate Phonetic	Word Error Rate Multi-sample Fenonic	Word Error Rate Combined
speaker 1	2.7%	1.7%	0.7%
speaker 2	1.0%	1.4%	0.8%
speaker 3	2.9%	2.5%	1.4%
speaker 4	3.2%	2.5%	1.4%
speaker 5	2.5%	2.2%	2.0%
speaker 6	6.3%	3.0%	2.0%
speaker 7	5.1%	2.7%	2.5%
speaker 8	4.1%	4.2%	3.6%
average of 8 speakers	3.5%	2.5%	1.8%

are identical except that *concerned* has an additional phone /d/ at the end. Consequently, for phonetic models, the only difference in the acoustic match probabilities for these two words will arise from the presence or absence of the /d/ at the end of the uttered word. The fenonic baseforms for these words are derived from 10 independent utterances of each word, and because of random variations in the utterances of the two words, the baseform for *concern* is not identical to the beginning part of baseform for *concerned.* Therefore, a new utterance of the word *concern* may sometimes be recognized as *concerned,* because the utterance matches the beginning of concerned better than it matches *concern.* The absence of the phone /d/ may not be sufficient to reduce the acoustic match score for *concerned* to below that for *concern.* No doubt, if the fenonic baseforms were constructed from many more sample utterances for each word, this random variation between similar words would disappear.

Noting the difference in the types of recognition errors for the two models, we explored the possibility of reducing the overall error rate by combining the results of both phonetic and fenonic acoustic matches in a single recognition system. This was achieved in the following way: Let $\Pr_P\{y|w\}$ and $\Pr_F\{y|w\}$ be the acoustic match probabilities obtained by matching the phonetic and fenonic models respectively, for the word w with the observed label string y. Then we define a combined match probability $\Pr_C\{y|w\}$ by

$$\Pr_C\{y|w\} = \Pr_P\{y|w\}^{\lambda} \cdot \Pr_F\{y|w\}^{(1-\lambda)} \tag{4}$$

where $0 \leq \lambda \leq 1$. The best value for λ was found empirically from recognition experiments with several speakers. We found the word error rate to be a slowly varying function of λ, with a fairly wide minimum at about $\lambda = 0.33$. This tuning of the value of λ was, of course, done on data from speakers other than those used in the recognition experiments reported in this paper. The rightmost column in Table III shows the recognition results for the combined acoustic match with λ set at 0.33. The combination of the phonetic and fenonic models reduced the average error rate to 1.8%, which is roughly a factor-of-2 improvement over the error rate with phonetic models alone.

V. CONCLUSIONS

We have described a method for constructing a new type of acoustic baseform for a word. This baseform is a sequence

of fenones. Fenones are very short acoustic events, and the inventory of fenones is obtained automatically by vector quantization of the acoustic space. Fenones are completely independent of any preconceived linguistic or phonetic notions. Fenonic baseforms model words at a much greater level of detail than phonetic baseforms, and result in a substantial reduction in the word error rate. Training of the fenonic Markov models for a new speaker can be accomplished with a short training script. We also describe a recognition system that uses both phonetic and fenonic models.

Fenonic baseforms are somewhat similar to word templates used in DP matching; with a crucial difference. The word models constructed from fenonic baseforms can be trained to a new speaker, whereas DP word templates cannot be trained.

There are, of course, many generalizations that have not been completely explored yet. The construction of baseforms from multiple acoustic strings was done in two steps. First, we established a set of fenones using singleton baseforms. Then we searched for the best baseform made up of these fenones. We can formulate the problem in a more general fashion, where the component fenones and the baseforms are estimated simultaneously. However, we have not yet been able to construct a reasonable algorithm to do this. Also, we have restricted the baseforms to be linear sequences of fenones. We are currently investigating the possibility of allowing more general graphs to be used in the baseforms. The decoding algorithm outlined in Section III, and described in detail in the Appendix, can be computationally very expensive in searching for the optimal baseform. We are investigating faster search methods for constructing baseforms. Nonetheless, since baseforms are constructed once and for all, the cost associated with baseform construction is only a one-time cost.

VI. Related Work

During the last few years, other researchers have also explored methods for the construction of full-word Markov models from automatically derived sub-word models. One such approach called the acoustic segment model was proposed by Lee, Juang, and Soong [24], and further work in this direction is described in articles by Lee *et al.* [25], Euler *et al.* [26], and Paliwal [27]. In this approach, words are first automatically segmented into sub-word pieces. Then the sub-word pieces are clustered to produce an inventory of acoustic segments. Acoustic models for words are then constructed by concatenating these acoustic segments.

The acoustic segment models ignore linguistic knowledge, as do the fenonic models discussed in this paper. The major difference between the two approaches is in the size of the sub-word units. Fenones represent acoustic events corresponding to a single frame. The acoustic segments of Lee, Juang, and Soong are typically several frames long, and their average length is roughly the same as the average duration of a phone. Like the fenonic models, the acoustic segment models also give better recognition results than phonetics-based models.

VII. Appendix

In this appendix we present a heuristic tree search algorithm for finding the best multi-sample fenonic baseform. An excellent discussion on heuristic tree search can be found in Chapter 3 of Nilsson [28]. Let us briefly review the basic principles of this technique. Heuristic tree search can be applied to problems in which the search space can be organized in the form of a tree. The terminal nodes of the tree correspond to complete solutions and the nonterminal nodes to partial solutions. An evaluation function is defined that measures the cost associated with a node. For terminal nodes, the evaluation function measures the total cost of the solution. For nonterminal nodes, the evaluation function combines the cost of the partial solution with a heuristic estimate of the additional cost to get from the nonterminal node to a terminal node. We start the search at the root node of the tree. A stack (or ordered list, in Nilsson's terminology) of nodes and their associated costs is maintained during the search process. Initially, the stack contains only the root node. At each iteration we find the best node in the stack. If the best node is a nonterminal node, we remove it from the stack, evaluate all its successor nodes, and add them to the stack. If the best node is a terminal node, we accept it as the solution and terminate the search. We now show how this search technique can be used for our problem.

Recall that we are given a set of fenones P, and a set of n label strings $y^{(1)}, y^{(2)}, \cdots, y^{(n)}$ corresponding to a word w. The optimal multi-sample fenonic baseform for w is defined to be the fenonic sequence $\hat{p} \in P^*$ such that

$$\hat{p} = \operatorname{argmax} \prod_{i=1}^{n} \Pr\{y^{(i)}|p\}. \tag{5}$$

First, we need to organize the search space in the form of a tree. During the search it will be important to distinguish between *complete* baseforms and *partial* baseforms. Complete baseforms account for all the labels in all the label strings, while partial baseforms leave at least some of the labels unaccounted for in some of the label strings. In order to distinguish between these two types of fenone sequences, we will terminate each complete baseform with the special symbol #. Let $P^*\#$ denote the set of all sequences of the form $p\#$, where $p \in P^*$. In this notation the sequences in $P^*\#$ correspond to all possible complete baseforms and the sequences in P^* to all possible partial baseforms. Now we represent all the strings in $P^* \cup P^*\#$ in the form of a tree. Each nonterminal node in the tree has $N_P + 1$ successors, one for each the N_P fenones and one for the distinguished terminal symbol #. There is an obvious correspondence between the fenone sequences and the nodes in the tree. The root node of the tree corresponds to the null fenone sequence, denoted by ϕ. At level m of the tree, there are $(N_P)^m$ nonterminal nodes corresponding to the $(N_P)^m$ distinct fenone sequences of length m in P^*, and $(N_P)^{m-1}$ terminal nodes corresponding to the $(N_P)^{m-1}$ distinct sequences of length m in $P^*\#$. The stack algorithm searches through this tree to find $\hat{p}\#$, the best terminal node, by starting at the root node and iteratively evaluating successors of the best nonterminal node found so far. When the best terminal node is found, the final # is stripped off, and the remaining fenone sequence is the baseform $\hat{p}$.

Now we need to define an appropriate evaluation function. For terminal nodes, the evaluation function should obviously

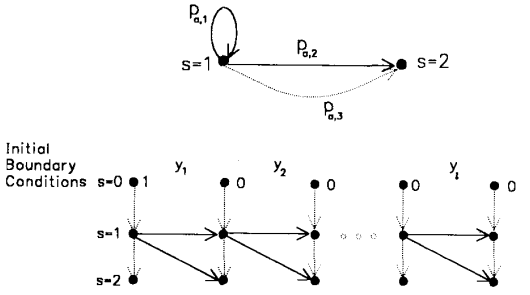

Fig. 5. Markov model and trellis for fenone P_a.

be the product of the acoustic match values for all the observed label strings, as expressed in eq. (2). For nonterminal nodes, the evaluation function should include an estimate of the cost of reaching a terminal node. We will first derive the evaluation function for the case when there is only one observed label string y, and then generalize it to multiple label strings.

First, let us briefly review the acoustic match calculation for Markov models. Fig. 5 shows the Markov model for a single fenone P_a. The model has two states, indexed as $s = 1$ and $s = 2$. Let $p_{a,1}, p_{a,2}$ and $p_{a,3}$ denote the self-loop, forward, and null transition probabilities for this model. Let $q_{a,k}$ and $r_{a,k}$ denote the output probabilities on the self-loop and forward transitions respectively. Note that a serves as an index in the fenone alphabet and k serves an index in the label alphabet. Let $y_{1\to l} = y_1, y_2, \cdots, y_l$, be an observed acoustic label string. Let $\alpha(s,t)$ denote the probability that the model generates the label string $y_{1\to t} = y_1, y_2, \cdots, y_t$, for $t \le l$, and is in state s. Starting with the boundary conditions

$$\alpha(0,0) = 1 \text{ and } \alpha(0,t) = 0 \text{ for } t > 0 \tag{6}$$

the acoustic match values $\alpha(s,t)$ can be calculated recursively for $t = 0, 1, 2, \cdots, l$ and $s = 1, 2$ using the following equations:

$$\alpha(1,t) = \alpha(0,t) + \alpha(1,t-1) \times p_{a,1} \times q_{a,y_t} \tag{7}$$

$$\alpha(2,t) = \alpha(1,t-1) \times p_{a,2} \times r_{a,y_t} + \alpha(1,t) \times p_{a,3} \tag{8}$$

This recursive calculation, which is generally called the forward-pass calculation, is depicted graphically in the trellis diagram of Fig. 5. The rows are indexed by the state index $s = 0, 1, 2$ and the columns by the frame index $t = 0, 1, \cdots, l$. The first row corresponding to $s = 0$ represents the initial boundary conditions of (6), and the next two rows represent (7) and (8).

We define the acoustic match vector for P_a as

$$\Lambda_{P_a} = (\lambda_{P_a}(0), \lambda_{P_a}(1), \cdots, \lambda_{P_a}(l)), \quad \text{where } \lambda_{P_a}(t) = \alpha(2,t). \tag{9}$$

Note that $\lambda_{P_a}(t)$ is the probability that P_a will produce the string $y_{1\to t}$. The rightmost value $\lambda_{P_a}(l)$ in this acoustic match vector corresponds to the total probability that the fenone P_a will produce the entire label string y. Thus the evaluation function for the complete baseform $P_a\#$ is given by

$$E(P_a\#) = \lambda_{P_a}(l). \tag{10}$$

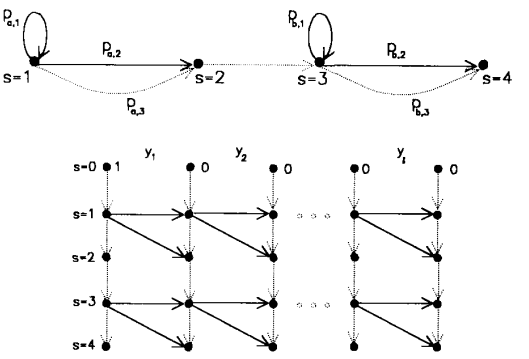

Fig. 6. Markov model and trellis for fenone sequence P_aP_b.

However, if we consider the partial baseform P_a, then P_a may only be responsible for an initial substring of y, and an extension of P_a will account for the remainder of y. Recall that $\lambda_{P_a}(t)$ is the probability that P_a will produce the initial substring $y_{1\to t}$. Let e denote the expected average value of the acoustic match per unit frame. (We will show later how to compute e.) Then $\lambda_{P_a}(t)e^{l-t}$ is the expected value for the acoustic match for the entire string $y_{1\to l}$, where P_a accounts for $y_{1\to t}$ and a continuation of it accounts for the remainder $y_{t+1\to l}$. Then for the partial baseform P_a we define the evaluation function to be

$$E(P_a) = \sum_{t=0}^{l} \lambda_{P_a}(t)e^{l-t}. \tag{11}$$

When we have multiple label strings $y^{(1)}, y^{(2)}, \cdots, y^{(n)}$, the total evaluation function is simply the product of the individual evaluation functions for each label string. If $E^{(i)}(P_a\#)$ and $E^{(i)}(P_a)$ denote the evaluation functions for the ith label string $y^{(i)}$, then the total evaluation function for all the label strings is given by

$$E(P_a\#) = \prod_{i=1}^{n} E^{(i)}(P_a\#) \tag{12}$$

$$E(P_a) = \prod_{i=1}^{n} E^{(i)}(P_a). \tag{13}$$

The value of e used in (11) is obtained as follows. Using the forward-pass algorithm we compute the total acoustic match value A for the entire training data using the singleton baseforms. If the training data has a total length of T frames, then we estimate e as

$$e = (A)^{1/T}. \tag{14}$$

In order to carry out the tree search efficiently, it is useful to be able to compute the evaluation function value for a successor node incrementally from its predecessor node. Fortunately, the evaluation function we use does have this desirable property. To see this, consider the acoustic match calculation for a fenone sequence P_aP_b which is an extension of the fenone sequence P_a. Fig. 6 shows the Markov model and the trellis

diagram for P_aP_b. The acoustic match equations in this case are

$$\alpha(1,t) = \alpha(0,t) + \alpha(1,t-1) \times p_{a,1} \times q_{a,y_t} \quad (15)$$

$$\alpha(2,t) = \alpha(1,t-1) \times p_{a,2} \times r_{a,y_t} + \alpha(1,t) \times p_{a,3} \quad (16)$$

$$\alpha(3,t) = \alpha(2,t) + \alpha(2,t-1) \times p_{b,1} \times q_{b,y_t} \quad (17)$$

$$\alpha(4,t) = \alpha(3,t-1) \times p_{b,2} \times r_{b,y_t} + \alpha(3,t) \times p_{b,3} \quad (18)$$

again with the boundary condition that $\alpha(0,0) = 1$ and $\alpha(0,t) = 0$ for $t > 0$. Notice that (15) and (16) are the same as (7) and (8), and that in order to evaluate (17) and (18) we need only the values in the acoustic match vector Λ_{P_a}. Thus, if we save the acoustic match vector Λ_{P_a} after computing the acoustic match for P_a, we can perform the acoustic match for the extended sequence P_aP_b without having to recompute the match for P_a. From the trellis diagram in Fig. 6, it is clear that the calculation of the first three rows is exactly the same as in Fig. 5. The calculation for the final two rows, which correspond to the states of fenone P_b, is similar to the calculation for fenone P_a in Fig. 5, except that the initial boundary conditions are replaced by the acoustic match vector of P_a.

Now, we can state the stack search algorithm for finding the best multi-sample baseform. A stack entry for a node in the tree will consist of the following:

- The fenone sequence corresponding to the node.
- The value of the evaluation function at the node.
- The acoustic match vectors for the node (one vector for each label string $y^{(i)}$).

The stack is initialized with the entry corresponding to the root node. The contents of this entry are as follows:

- The null fenone sequence ϕ.
- The evaluation function value $E(\phi) = e^{l_1} \times e^{l_2} \times \cdots e^{l_n}$.
- The acoustic match vectors $\Lambda_\phi^{(1)}, \Lambda_\phi^{(2)}, \cdots, \Lambda_\phi^{(n)}$. The acoustic match vectors for the root node correspond to the boundary condition in (6). Thus $\Lambda_\phi^{(i)}$ is of length $l_i + 1$ and is of the form $(1, 0, 0, \cdots, 0)$.

The search algorithm is as follows:

1. Initialize.
   Insert root node entry in stack.
2. Find best entry.
   Find the entry in the stack with the highest value of the evaluation function.
   If the best entry is a nonterminal node, go to step 3.
   Otherwise, go to step 4.
3. Extend.
   Remove best entry from stack.
   Construct the $N_P + 1$ successors of this entry.
   Insert each successor in the stack.
   Go to step 2.
4. Terminate.
   Strip the final # from the fenone sequence of the best entry and accept it as the baseform.
   Stop.

Nilsson [28] proves that this heuristic search method is guaranteed to find the optimal solution only if the heuristic cost function satisfies a certain condition. In our case, the necessary condition is that the estimated match value for the string $y_{t \to l}$ be an overestimate of the actual match for that string. In our search, the estimated value e^{l-t} is not necessarily larger than the true value, thus finding the optimal baseform is not guaranteed. We can increase the likelihood of finding the optimal baseform by increasing the value of e. This unfortunately also increases the amount of search, which is not desirable. In practice we have found that letting e be the average match per unit frame on the training data works very well.

ACKNOWLEDGMENT

We would like to thank Seppo Haltsonen for carrying out the template-based DP matching experiments. We would also like to thank the members of the Speech Recognition group at the IBM Research Center, without whose cooperation this work would not have been possible.

REFERENCES

[1] A. Averbuch, L. Bahl, R. Bakis, P. Brown, A. Cole, G. Daggett, S. Das, K. Davies, S. De Gennaro, P. de Souza, E. Epstein, D. Fraleigh, F. Jelinek, S. Katz, B. Lewis, R. Mercer, A. Nadas, D. Nahamoo, M. Picheny, G. Shichman, and P. Spinelli, "An IBM-PC based large-vocabulary isolated-utterance speech recognizer," in *Proc. 1986 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* Tokyo, Japan, pp. 53–56, Apr. 1986.

[2] A. Averbuch, L. Bahl, R. Bakis, P. Brown, G. Daggett, S. Das, K. Davies, S. De Gennaro, P. de Souza, E. Epstein, D. Fraleigh, F. Jelinek, B. Lewis, R. Mercer, J. Moorhead, A. Nadas, D. Nahamoo, M. Picheny, G. Shichman, P. Spinelli, D. Van Compernolle, and H. Wilkens, "Experiments with the TANGORA 20,000 word speech recognizer," in *Proc. 1987 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* Dallas, TX, pp. 701–704, Apr. 1987.

[3] T. K. Vintsyuk, "Element-wise recognition of continuous speech composed of words from a specified dictionary," *Kibernetika,* vol. 7, pp. 133–143, Mar.–Apr. 1971.

[4] R. Bakis, "Continuous speech recognition via centisecond acoustic states," presented at the 91st Meeting Acoustical Society of America, Washington, DC, Apr. 1976.

[5] F. Jelinek, "Continuous speech recognition by statistical methods," in *Proc. IEEE,* vol. 64, pp. 532–556, Apr. 1976.

[6] L. E. Baum and T. Petrie, "Statistical inference for probabilistic functions of finite state Markov chains," *Annals of Math. Statist.,* vol. 36, pp. 1554–1563, Dec. 1966.

[7] L. E. Baum, "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes," *Inequalities,* vol. 3, pp. 1–8, 1972.

[8] L. R. Rabiner and S. E. Levinson, "A speaker-independent, syntax-directed, connected word recognition system based on hidden Markov models and level-building," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-33, pp. 561–573, June 1985.

[9] F. Jelinek, L. R. Bahl, and R. L. Mercer, "Design of a linguistic decoder for the recognition of continuous speech," *IEEE Trans. Information Theory,* vol. IT-21, pp. 250–256, May 1975.

[10] L. R. Bahl and F. Jelinek, "Decoding for channels with insertions, deletions, and substitutions with applications to speech recognition," *IEEE Trans. Information Theory,* vol. IT-21, pp. 404–411, July 1975.

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

[12] P. S. Cohen and R. L. Mercer, "The phonological component of an automatic speech-recognition system," pp. 275–320 in *Speech Recognition,* D. R. Reddy, Ed. New York: Academic Press, 1975.

[13] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Recognition results with several experimental

acoustic processors," in *Proc. 1979 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* Washington, DC, pp. 249–251, Apr. 1979.
[14] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Further results on the recognition of a continuously read natural corpus," in *Proc. 1980 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* Denver, CO, pp. 872–875, Apr. 1980.
[15] R. Schwartz, Y. Chow, S. Roucos, M. Krasner, and J. Makhoul, "Improved hidden Markov modeling of phonemes for continuous speech," in *Proc. 1984 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* San Diego, CA, pp. 35.6.1–35.6.4, Mar. 1984.
[16] K. F. Lee, "Context-dependent phonetic hidden Markov models for speaker independent continuous speech recognition," *IEEE Trans. on Acoustics, Speech, and Signal Processing,* vol. ASSP-38, pp. 599–609, Apr. 1990.
[17] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.,* vol. 1, pp. 4–29, Apr. 1984.
[18] J. Makhoul, S. Roucos, and H. Gish, "Vector quantization in speech coding," in *Proc. IEEE,* vol. 73, pp. 1551–1588, Nov. 1985.
[19] J. Cohen, "Application of an auditory model to speech recognition," *J. Acoust. Soc. Amer.,* vol. 85, pp. 2623–2629, June 1989.
[20] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-23, pp. 67–72, Feb. 1975.
[21] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-26, pp. 194–200, Feb. 1978.
[22] S. Haltsonen, "Improved dynamic time warping methods for discrete utterance recognition," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-33, pp. 449–450, Apr. 1985.
[23] S. Haltsonen, "Recognition of isolated word sentences from a large vocabulary using dynamic time warping methods," *IEEE Trans. Acoustics, Speech, and Signal Processing,* vol. ASSP-33, pp. 1026–1027, Aug. 1985.
[24] C. H. Lee, B. H. Juang, and F. K. Soong, "A segment model based approach to speech recognition," *Proc. 1988 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* New York, NY, pp. 501–504, Apr. 1988.
[25] C. H. Lee, B. H. Juang, F. K. Soong, and L. R. Rabiner, "Word recognition using whole word and subword models," in *Proc. 1989 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* Glasgow, UK, pp. 683–686, May 1989.
[26] S. Euler, B. H. Juang, C. H. Lee, and F. K. Soong, "Statistical segmentation and word modeling techniques in isolated word recognition," in *Proc. 1990 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* Albuquerque, NM, pp. 745–748, Apr. 1990.
[27] K. K. Paliwal, "Lexicon-building method for an acoustic sub-word based speech recognizer," in *Proc. 1990 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing,* Albuquerque, NM, pp. 729–732, Apr. 1990.
[28] N. J. Nilsson, *Problem-solving Methods in Artificial Intelligence.* New York: McGraw-Hill, 1971.

Lalit R. Bahl (S'66–M'68), for a photograph and biography please see page 67 of the January issue of this TRANSACTIONS.

Peter F. Brown, for a photograph and biography please see page 83 of the January issue of this TRANSACTIONS.

Peter V. de Souza, for a photograph and biography please see page 83 of the January 1993 issue of this TRANSACTIONS.

Robert L. Mercer (M'83), for a photograph and biography please see page 67 of the January 1993 issue of this TRANSACTIONS.

Michael A. Picheny (S'73–M'80), for a photograph and biography please see page 344 of the July issue of this TRANSACTIONS.