Isolated-Word Speech Recognition Using Multisection Vector Quantization Codebooks

DAVID K. BURTON, MEMBER, IEEE, JOHN E. SHORE, SENIOR MEMBER, IEEE,
AND JOSEPH T. BUCK, MEMBER, IEEE

***Abstract*—A new approach to isolated-word speech recognition using vector quantization (VQ) is examined. In this approach, words are recognized by means of sequences of VQ codebooks, called multisection codebooks. A separate multisection codebook is designed for each word in the recognition vocabulary by dividing the word into equal-length sections and designing a standard VQ codebook for each section. Unknown words are classified by dividing them into corresponding sections, encoding them with the multisection codebooks, and finding the multisection codebook that yields the smallest average distortion. For speaker-independent recognition of the digits, this approach achieved a recognition accuracy of 98 percent. In addition, the approach achieved greater than 99 percent accuracy for speaker-dependent recognition of the digits with only one distortion computation per input frame per vocabulary word. The approach is described, detailed experimental results are presented and discussed, and computational requirements are analyzed.**

I. Introduction

VECTOR quantization (VQ) is a data compression principle [1], [2] with several successful applications, including speech coding, [3]–[5] image coding [6], [7], and speech recognition [8]–[19]. In previous work on speech recognition [9], [10], [17], we developed a method in which isolated words are classified by means of the average distortion that results from encoding them with VQ codebooks. In this paper, we present a generalization of that method. The generalization, which improves recognition performance and reduces computational requirements, was motivated by work of Buzo *et al.* [11].

In our previous approach [17], a VQ codebook is generated for each word in the recognition vocabulary by applying an information-theoretic iterative clustering technique [20] to a training sequence containing several repetitions of the vocabulary word. This clustering process removes all time-sequence information from the training sequence and represents each vocabulary word as a set of independent spectra. An input utterance is classified by encoding it with every codebook and finding the codebook that yields the smallest average distortion. Because the average distortion does not depend on the sequence of input speech frames, this approach performs isolated-word recognition entirely without time alignment.

With just four spectra in each codebook, our previous approach achieved 97.7 percent accuracy for speaker-dependent recognition of a twenty-word vocabulary [17]. With eight spectra in each codebook, the accuracy increased to 98.8 percent [17]. These results showed that much more can be done without time-sequence information than is commonly assumed. For suitably chosen vocabularies, characteristic spectra contain enough information for recognition, and information theoretic clustering does a good job of extracting that information from training data.

To improve recognition performance and to decrease computational complexity, we have been investigating ways of incorporating time-sequence information into the recognition procedure. Here, we present results for a new method that incorporates time-sequence information by means of sequences of VQ codebooks that we refer to collectively as *multisection* codebooks. A separate multisection codebook is designed for each word in the recognition vocabulary by dividing the words in the codebook's training sequence into equal-length sections and designing a standard VQ codebook for each section. Unknown words are classified by dividing them into appropriate sections, performing VQ on a section by section basis, and finding the multisection codebook that yields the smallest average distortion. The new approach reduces to our previous approach when the number of sections is reduced to one. Henceforth, we refer to our previous approach as the *single-section* case. Preliminary results for the multisection approach were reported in [13], [18].

VQ has also been used by others to reduce the computational and memory requirements of existing isolated-word recognition approaches [8], [12], [14], [15], [16], [19]. In these approaches, spectra from a single, large VQ codebook are used to replace the spectra of both input speech frames and stored reference data. Our approach is quite different, both because we design separate codebooks for each word in the recognition vocabulary, and because we avoid standard methods of time alignment.

After explaining our speech recognition approach in Section II, we describe the databases and experiments in Section III. Section IV contains the results for speaker-independent recognition, and Section V contains results for speaker-dependent recognition. We discuss computational considerations in Section VI, and we present some general conclusions in Section VII.

Manuscript received March 26, 1984; revised October 18, 1984.

The authors are with the Computer Science and Systems Branch, Information Technology Division, Naval Research Laboratory, Washington, DC 20375.

0096-3518/85/0800-0837$01.00 © 1985 IEEE

II. Approach

In this section, we give background information and describe the multisection approach.

A. Vector Quantization

VQ is an information-theoretic data compression principle introduced by Shannon in the late 1950's [21]. For a specified transmission rate, the objective of VQ is to find the set of reproduction vectors, or codebook, that represents an information source with minimum expected "distortion." The data compression is achieved by transmitting a reproduction vector index rather than the original source vector. In general, the selection of a perceptually meaningful distortion measure and the construction of an optimal codebook are difficult problems. For speech, however, good choices exist [3], [4].

Speech coding by VQ is a narrow-bandwidth speech coding technique based on linear predictive coding (LPC) [3], [4]. The shape of the speech spectrum in each 20 ms or so time interval (frame) is encoded as the index of a prestored set of LPC parameters that define an autoregressive model and is called a *codeword*. The collection of codewords is called a *codebook*. Let $\boldsymbol{C} = \{C_1, C_2, \cdots, C_N\}$ be a codebook of N codewords C_i, and let S_j be the autocorrelation estimates from the jth frame of the speech to be coded. Then the spectrum shape of the jth frame is coded by identifying the codeword C_b that "best represents" S_j according to the "nearest-neighbor rule"

$$d(S_j, C_b) = \min_i d(S_j, C_i), \tag{1}$$

for some distortion measure d.

Vector quantization codebooks are designed to minimize the average distortion that results from encoding a long training sequence of speech frames. In particular, if $T_j, j = 1, \cdots, L$ is such a training sequence, the codebook $\boldsymbol{C}$ is designed so that

$$\frac{1}{L}\sum_{j=1}^{L} \min_i d(T_j, C_i) \tag{2}$$

achieves at least a local minimum. The codebook design algorithm used here is based on the work in [20] and [3]. Put simply, the L frames of the training sequence are divided into N clusters such that all the frames in a cluster have similar spectrum shapes. The N codewords are the centroids of these clusters.

B. Isolated-Word Recognition

In speech coding by VQ, a single codebook is designed from a long training sequence that is representative of all speech to be encoded by the system. In the single-section approach to isolated-word recognition [9], [10], [17], we used a separate codebook for each word in the recognition vocabulary. Our new method, based on [11], represents each vocabulary word as a time-dependent sequence of section codebooks, which we call a multisection codebook. New words are classified by performing VQ and finding the multisection codebook that achieves the smallest average distortion.

To be more precise, let V be the number of words in the recognition vocabulary, and let T_k be the number of utterances in the training sequence used to design codebook $\boldsymbol{C}_k$ for the kth vocabulary word, where $k = 1, \cdots, V$. Also, let F_{qk} be the number of frames in the qth utterance in the training sequence for $\boldsymbol{C}_k$, where $q = 1, \cdots, T_k$, and finally, let U_{mqk} be the mth frame in the qth training utterance for $\boldsymbol{C}_k$, where $m = 1, \cdots F_{qk}$. Then there are V multisection codebooks $\boldsymbol{C}_k$, each comprising a sequence of VQ *section codebooks* $\boldsymbol{C}_{kj}$. The section codebook $\boldsymbol{C}_{kj}$ is designed using n frames from each training utterance for the kth vocabulary word. That is, $\boldsymbol{C}_{kj}$ is designed from the frames U_{mqk}, where $m = (j-1)n + 1, \cdots, jn$, and $q = 1, \cdots T_k$. In particular, $\boldsymbol{C}_{k1}$ is designed from the first n frames of each training utterance for the kth word in the recognition vocabulary, $\boldsymbol{C}_{k2}$ from the second n frames, etc. We call n the *compression factor*—it is the number of frames that are spanned per section. If, for a particular training utterance q, m is greater than F_{qk}, the corresponding frames U_{mqk} lie beyond the end of the word and are not included in the training sequence for $\boldsymbol{C}_{kj}$. Finally, let C_{kji}, $i = 1, \cdots, N_{kj}$ be codewords in section codebook C_{kj}. We call the V multisection codebooks $\{\boldsymbol{C}_k; k = 1, \cdots, V\}$ a *codebook set*.

Suppose a new utterance to be classified contains L frames, and P_l is the set of autocorrelation estimates from the lth frame ($l = 1, \cdots, L$). Now let D_k be the *average distortion* resulting from coding the unknown utterance with the codebook $\boldsymbol{C}_k$,

$$D_k = \frac{1}{L}\sum_{j=1}^{S_k} d_{kj} \tag{3}$$

where S_k is the number of section codebooks in $\boldsymbol{C}_k$, and

$$d_{kj} = \sum_{l=(j-1)n+1}^{\min[jn, L]} \min_i d(P_l, C_{kji}) \tag{4}$$

is the total distortion from coding the jth section of the input with the jth section codebook $\boldsymbol{C}_{kj}$ of $\boldsymbol{C}_k$, and where n is the compression factor. Then the utterance is classified as the rth word in the recognition vocabulary, where

$$D_r = \min_k D_k.$$

If desired, one can select a set of threshold values $D_{\min}$ and require $D_r < D_{\min}$ in (3) for a valid classification. This can improve classification reliability.

If, in the above description, all words are aligned at their beginnings, we call the approach *left-aligned*. In the left-aligned case, variations in speaking rates often result in several sounds being included in the training sequences for individual section codebooks. To reduce this effect, we also tried linearly normalizing all training sequences and classification utterances to the same length. We call this approach *length-normalized*.

In the length-normalized approach, the number of sections in the input word is always equal to the number of section codebooks. In the left-aligned approach, however, the input word can have more or less sections than the codebooks; we stop encoding a word in a codebook when we run out of either input word frames or codebook sections.

In the foregoing terms, the approach in [11] corresponds to left-alignment with $n = 1$. For left-alignment with n greater than or equal to the maximum number of frames in all the training utterances, the multisection approach reduces to our previous single-section approach [9], [10], [17].

C. Multisection Codebooks

Each classification codebook $\boldsymbol{C}_k$ is designed from a separate training sequence containing repetitions of the kth word in the recognition vocabulary. A speaker-dependent codebook is made from a training sequence spoken by one person. The resulting codebooks are then used to classify additional utterances from that speaker. For speaker-independent codebooks, the training sequence for each codebook is spoken by several people and the codebooks are used to classify additional utterances from people other than those who contribute to the training sequence.

We used three types of multisection codebooks:

1) fixed-size codebooks,
2) fixed-distortion codebooks,
3) unclustered codebooks.

The three codebook types are further discussed below.

As the name implies, in a *fixed-size* codebook the section codebook size N_{kj} is specified ahead of time and the design algorithm chooses N_{kj} codewords that minimize the average distortion resulting from encoding the training sequence for a particular section codebook. Section codebook sizes are limited for convenience to powers of 2, i.e., $N_{kj} = 2r_k$, where r_k is called the *rate* of C_{kj}. All section codebooks (and thus multisection codebooks) in a fixed-size codebook set have the same number of code words.

For a *fixed-distortion* codebook, the design algorithm increases the section codebook size until it can design a section codebook that encodes the training sequence with an average distortion that is less than or equal to a prespecified value T. All section codebooks in a fixed-distortion codebook set are generated with the same average distortion threshold and can therefore have different sizes. Like fixed-size section codebooks, the size of fixed-distortion section codebook are limited to powers of 2.

The third type of codebook is the *unclustered codebook*. These are generated without the clustering algorithm, simply by making a codeword out of each frame in the training sequence. Our motivation for considering unclustered codebooks was two-fold. The first was computational efficiency and convenience—generating them is much easier than generating clustered codebooks. The second was as a measure of performance. Since the clustering procedure attempts to find spectrum shapes that are representative of the training sequence, the effectiveness of clustering can be evaluated by comparing the performance of clustered and unclustered codebooks designed from the same training sequence.

D. Distortion Measures

In generating codebooks for voice coding, two distortion measures are effective [3], [22]: the *Itakura–Saito* (d_{IS}) and the *gain normalized Itakura–Saito* (d_{GN}). Of the two, d_{GN} codebooks are better for isolated word recognition than d_{IS} codebooks, particularly when using small codebooks built from short training sequences [17]. Thus, we used d_{GN} codebooks in the work reported herein.

For the classification distortion measure in (4), we used the *gain optimized Itakura–Saito* distortion measure, which was popularized by Itakura [23] and is also known as the log-likelihood distortion measure. Properties of all three distortion measures are discussed in [24].

E. LPC Parameters

LPC parameters for both codebook generation and utterance classification were generated using the autocorrelation method with Hamming windowing. Except for N, the number of points to shift between successive speech frames, we chose analysis conditions for compatibility with the Navy's 2.4 kbit/s LPC-10 system [25]: analysis window width = 130 points, filter order = 10, and preemphasis = 94 percent. When using the length-normalized approach, N was adjusted to satisfy the normalization length requirement; however, when using the left-aligned approach, $N = 180$ was used as is done for the Navy's LPC-10 system. The LPC analysis parameters used in classifications were always chosen to match those used in generating the codebooks.

III. Experimental Background

Our experiments were conducted using two databases that were prepared by Texas Instruments, Inc. (TI). The first, which we call TI-1, is a 16-speaker database containing a 20-word vocabulary [26]. The second, which we call TI-2, is a 300-speaker database containing the digits spoken in isolation and as connected strings [27]. We first tuned our algorithm based on prior experience and on a speaker-independent male-only parameter study using TI-1. Using the isolated digits, we then tested the tuned algorithm on the male and female speakers in TI-2. In addition, we tested the tuned algorithm in a speaker-dependent mode using TI-1.

Automatic endpoint detection for both training-sequence and classification utterances was used in our experiments. Our endpoint-detection algorithm is based on ideas presented in [28], [29], and is described in [17]. Briefly, the algorithm first analyzes the background noise to determine its average magnitude and then uses the results to set various thresholds that are used to find significant "energy clumps" in the data.

In the rest of this section we describe the database, the experimental parameters, and the experiments.

A. TI Databases

The database TI-1 consists of twenty words: the digits *zero* through *nine* and the ten control words *yes, no, erase, rubout, repeat, go, enter, help, stop,* and *start* [26]. Eight male and eight female speakers each recorded twenty-six repetitions of each word in the vocabulary, for a total of 8320 utterances. The data was recorded on analog tape under tightly controlled conditions: the noise level was low, the speech level was restricted to a ±3 dB range, the acoustic environment was unvarying, and all errors in the input words were eliminated. After collection, the data was low-pass filtered and sampled at 12 500 samples per second. We received the data in digital form on magnetic tape. Each utterance, preceded and followed by short segments of ambient noise, was contained in a separate file. In a previous study using single-section codebooks [17], we used the data primarily at the 12 500 sampling rate. For the work reported here, the data was down sampled to 8000 samples per second. The down sampling procedure is described in [17].

The database TI-2 contains data from 111 adult males, 114 adult females, and 101 children (50 boys and 51 girls), and it is divided into two separate parts: a training portion and a testing portion [27]. The training portion contains data from 55 males, 57 females, and 51 children; the testing portion contains data from the rest of the speakers. The data was collected in an acoustically treated sound room and digitized at 20 000 samples per second using a 16-bit A/D converter. As was the case with TI-1, all errors in the input words were eliminated. We received the data after it had been down sampled to 8000 samples per second. Although the database contains the 10 digits (*zero* through *nine*) and the word "*oh*" for all the speakers, for compatibility with other work we used only the isolated digits spoken by the adults. The database contains two utterances of each digit by each speaker for a total of 2240 training and 2260 test digits.

B. Experimental Parameters

In this subsection, we describe the experimental parameters associated with codebook generation and utterance classification. The codebook generation parameters are as follows:

1) number of utterances in the training sequence;

2) energy threshold $E_{\min}$, where E is computed by

$$E = \sum_{i=1}^{W} x_i^2,$$

here, W is the analysis window width, and x_i are the time-domain samples after preemphasis and Hamming windowing;

3) left-alignment or length-normalized alignment;

4) compression factor; and

5) codebook type and size.

The energy threshold is used to ignore nearly-silent frames; frames with energy below this threshold are not used in designing codebooks or performing a classification. For all the work reported here, we used $E_{\min} = 250$.

The parameters associated with utterance classification are as follows:

1) compression factor;

2) utterance alignment; and

3) energy threshold.

For consistency these values were chosen to match those used in the codebook generation.

C. List of Experiments

In this subsection, we list the experiments reported in the remainder of the paper. Section IV contains the results of the following speaker-independent parameter studies and experiments, listed according to the corresponding subsection of Section IV:

IV-A) A male-only study of recognition accuracy as a function of compression factor and section codebook rate (TI-1).

Comparison of recognition performance using unclustered and clustered codebooks when using the "best" compression factor (TI-1).

Study of recognition accuracy as a function of the normalization length (TI-1).

Recognition accuracy comparison using fixed-size and fixed-distortion codebook sets (TI-1).

Recognition accuracy comparison of left-aligned and length-normalized approaches (TI-1).

IV-B) A male-only experiment using the adult speakers in TI-2.

IV-C) A female-only experiment using the adult speakers in TI-2.

IV-D) A combined male-and-female experiment using the adult speakers in TI-2.

Section V contains the results of speaker-dependent experiments. The experiments are listed according to the corresponding subsection of Section V:

V-A) Comparison of multisection and single-section recognition performance on TI-1.

V-B) A rate-0 multisection study (TI-1).

V-C) Recognition results for fixed-size codebooks with short training sequences (TI-1).

Recognition results for unclustered codebooks with short training sequences (TI-1).

IV. Speaker-Independent Experiments

In this section, we describe four sets of experiments. The first set were parameter studies done on the male speakers in TI-1—we varied the compression factor, section codebook rate, utterance alignment, and codebook design method. Based on the results, we give guidelines for parameter selection. In the rest of the experiments, the parameters were fixed based on the results of the first set of experiments. We tested speaker-independent classification of the digits using the adult male speakers in TI-2. We did the same for the female speakers in TI-2, and we tested speaker-independent classification of the digits using all the adult speakers (male and female) in TI-2.

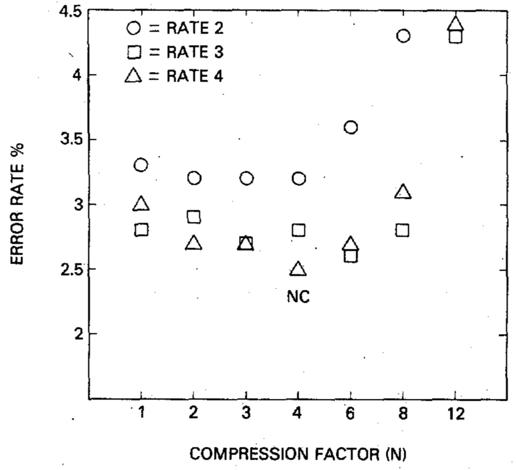


Fig. 1. Relationship among compression factor, error rate, and section codebook rate for speaker-independent recognition.

A. Male Parameter Study

For all parameter studies, the LPC parameters are those specified in Section II-E. We considered each of the 8 male speakers in turn. For each male speaker, we classified 520 utterances using codebooks designed from the first 9 utterances from each of the other 7 males.

In the first parameter study we examined the relationships among compression factor, section codebook rate, and recognition accuracy. We used a 24-frame length-normalized approach—24 frames was about the average length of the words in the recognition vocabulary. We used fixed-size section codebooks with rates 2, 3, and 4 together with compression factors 1, 2, 3, 4, 6, 8, and 12. The results are plotted in Fig. 1. Note that each point on the plot represents 4160 speaker-independent classifications—520 classifications per speaker for 8 speakers.

Based on Fig. 1, we make the following observations:

1) at each compression factor, the error spread is less than 2 percent for all section codebook rates;

2) the difference in error rates between section codebook rates 2 and 3 is generally small, but it is consistent and significant;

3) there is no significant difference in error rates for section codebook rates 3 and 4;

4) a compression factor between 3 and 6 appears best.

To gain insight into any relationship among word complexity (such as the number of syllables or phonemes), compression factor, and error rate, we examined the number of errors as a function of compression factor for the nondigit words. We had conjectured that simpler words like *no, go,* and *yes* would be easier to recognize using larger compression factors, and that more complex words like *repeat, rubout,* and *start* would require smaller compression factors. The data, however, showed no obvious correlation between word complexity, error rate, and compression factor.

Previously [17], we performed a similar speaker-independent classification experiment on these same 8 male speakers. There we used the single-section approach and the original 12 500 samples per second data. The training method was the same as used here: nine utterances from each of the seven speakers not being classified were used to build codebooks. The analysis conditions consisted of the following: $N = 250$ (20 ms), analysis window $= 250$ points, filter order $= 16$, preemphasis $= 90$ percent, and Hamming windowing. As in this study, the autocorrelation method of LPC was used. Using rate-5, single-section codebooks, an average recognition accuracy of 88 percent was achieved, as opposed to the 97.5 percent achieved by the current approach. Thus by using the multisection approach, the number of distortion computations per classification has been reduced (by a factor of 4 for rate-3 section codebooks), and the number of errors has been reduced by about a factor of 4.

TABLE I
MALE SPEAKER-INDEPENDENT RECOGNITION: LENGTH-NORMALIZATION STUDY

		Length = 12 Frames	Length = 24 Frames	Length = 36 Frames
Speaker	No. Class.	% Correct	% Correct	% Correct
TBS	520	97.2	97.7	97.3
WMF	520	94.0	96.7	97.9
RLD	520	95.6	95.4	96.4
GRD	520	93.1	95.8	95.8
KAB	520	95.8	96.0	95.2
MSW	520	98.5	98.1	98.7
REH	520	98.5	98.9	99.0
RGL	520	97.9	99.4	99.8
all	4160	96.3	97.2	97.5

As stated earlier, unclustered codebooks are generated by making a codeword out of each frame in the training sequence, and the effectiveness of clustering can be evaluated by comparing the performance of unclustered and clustered codebooks designed from the same training sequence. We built unclustered codebooks using a compression factor of 4 and the same LPC analysis parameter as specified for the clustered codebooks. The result is marked by *NC* in Fig. 1. The degradation in recognition performance using rate-3 clustered codebooks instead of unclustered codebooks is small—about 0.5 percent. Since the rate-3, multisection codebooks are only about $\frac{1}{30}$ the size of the unclustered codebooks and the error rates for the two are close, it is apparent that the clustering procedure performs an effective data compression function.

Next we studied the effect of normalization length on recognition accuracy. We felt that, in general, longer normalization lengths would result in higher recognition accuracies. Doubling the normalization length, however, also doubles the number of distortion computations needed to compare an input utterance with a codebook. We were searching for the shortest normalization length that did not significantly degrade the recognition accuracy. To study this, we chose normalization lengths of 12, 24, and 36. We used rate-3 section codebooks, and the compression factor was adjusted in each case so that there were 6 section codebooks per word. Note that for a fixed analysis window width, increasing normalization length increases the overlap between adjacent analysis frames.

The results, listed in Table I, show that the average recognition accuracy increases gradually with increases in the normalization length. The question remains, however, whether the increase is significant.

To test for statistical significance, we used the two-sample Wilcoxon rank sum test [30]. For this test, let $F(x)$ be the probability distribution function describing the recognition accuracy x of a multisection approach with a specific set of multisection parameters (compression factor, section codebook rate, normalization length, etc.). In the normalization length study described above, let $F_s(x)$ be the probability distribution function describing the recognition performance of one of the shorter length-normalized approachs, and let $F_l(x)$ be the probability distribution function for an approach with a longer normalization length. Also, let μ_s be the mean recognition accuracy corresponding to $F_s(x)$, and let $F_l(x)$ have a mean μ_l. The null hypothesis for our test is $F_s(x) = F_l(x)$ for all x: thus, $\mu_s = \mu_l$. The alternative hypothesis is $F_s(x) = F_l(x + \Delta)$ for some positive Δ, or $F_s(x)$ is shifted to the left of $F_l(x)$. This implies $\mu_s < \mu_l$.

We performed the Wilcoxon test for all three combinations: 12 versus 24, 12 versus 36, and 24 versus 36. The significance levels for rejection of the null hypothesis of equal mean recognition accuracies were 0.186, 0.104, and 0.397, respectively. Based on the Wilcoxon test results and the average recognition accuracies, we believe the increase in computations in going from 12 frames to 24 frames is justified, but the increase in going to 36 frames is not justified. (These conclusions about normalization length are similar to ones obtained by Dautrich *et al.* for isolated-word recognition approaches that use dynamic time warping to do time alignment [31].)

Previously [17], we compared the performance of fixed-distortion and fixed-size codebooks using the single-section approach. Although in that study the fixed-size codebooks performed better than the fixed-distortion codebooks, we felt this might not hold true when using multisection codebooks. One reason is that each section codebook represents only a small portion of a word instead of the whole word as in the single section approach. This restriction might reduce the types of confusions that earlier caused fixed-distortion codebooks to perform worse than fixed-size codebooks. The possible advantages of fixed-distortion codebooks are that each fixed-distortion codebook is only as large as necessary to satisfy the distortion criterion. Thus it follows that fixed-distortion codebooks might lead to the same classification performance as fixed-size codebooks but with fewer total codewords. This could lead to smaller memory requirements and faster classification performance.

We chose $T = 0.45$ and $T = 0.30$ as distortion thresholds, and we designed fixed-distortion codebooks sets using the same conditions as used in the previous fixed-size codebook studies. For the $T = 0.45$ threshold, the average section codebook size was 7.35 codewords; for the $T = 0.30$ threshold, it was 15.99 codewords.

The average recognition accuracy using the fixed-distortion codebooks with $T = 0.45$ was 96.5 percent. With $T = 0.30$, the recognition accuracy was 96.8 percent. The fixed-size, rate-3 and -4 codebook sets had recognition accuracies of 97.2 and 97.5 percent, respectively. So, as before [17], the fixed-size codebooks discriminate better in word recognition than do fixed-distortion codebooks.

TABLE II
MALE SPEAKER-INDEPENDENT RECOGNITION: LEFT-ALIGNED VERSUS LENGTH-NORMALIZED CODEBOOKS

Speaker	No. Class.	Left Aligned		Length Normalized	
		Errors	% Correct	Errors	% Correct
WMF	520	34	93.5	17	96.7
RLD	520	21	96.0	24	95.4
RGL	520	14	97.3	3	99.4
MSW	520	22	95.8	10	98.1
GRD	520	25	95.2	22	95.8
TBS	520	20	96.2	12	97.7
KAB	520	34	93.5	21	96.0
REH	520	25	95.2	6	98.9
all	4160	195	95.3	115	97.2

So far, the experiments used length-normalized codebooks. We tested the left-aligned approach using a compression factor of 4, a section codebook rate of 3, and, except for N (the number of points to shift between successive speech frames), the same analysis conditions as before. In the left-aligned experiment, N was fixed at 180. Left-alignment was used both to design codebooks and to classify input utterances.

The left-aligned results together with the rate-3, compression factor 4, length-normalized results are shown in Table II. The length-normalized approach is clearly superior. This conclusion is also supported by the Wilcoxon test: the significance level is 0.012 for rejecting the null hypothesis of equal mean recognition accuracies.

The foregoing results suggest the following guidelines:

1) length normalization should be used with analysis conditions that provide frame overlap;

2) the compression factor should correspond to roughly 20 percent of the normalized length;

3) fixed-size section codebooks of at least rate-3 should be used.

Although the speakers in these studies possessed several of the major dialects, the speaker sample was small and homogeneous—8 male speakers living in Texas. Thus, the rate-3 section codebooks might be too small. In the next three sections we further evaluate this issue by studying the performance of the method on a larger population database.

B. Male Results

Using a compression factor of 4 and 24-frame length normalization, we studied speaker-independent recognition using the 111 male speakers in TI-2. We used two utterances of each digit from each of the 55 training speakers to build codebooks, and we classified two utterances of each digit from each of the 56 test speakers. Table III contains the results for section rates 1 through 4. The results using rate-3 and -4 section codebooks were the same (98.8 percent correct), but the actual errors were a little different. For both rate-3 and -4 section codebooks, however, the most common errors were for the words *six* and *eight*—for both rates, there were 4 errors on *six* and 4 errors on *eight*.

TABLE III
MALE SPEAKER-INDEPENDENT RECOGNITION OF THE DIGITS: SECTION RATES 1 THROUGH 4

Section Rate	No. Class.	Errors	% Correct
1	1120	31	97.2
2	1120	23	98.0
3	1120	14	98.8
4	1120	14	98.8

TABLE IV
FEMALE SPEAKER-INDEPENDENT RECOGNITION OF THE DIGITS: SECTION RATES 1 THROUGH 4

Section Rate	No. Class.	Errors	% Correct
1	1140	31	97.0
2	1140	26	97.7
3	1140	24	97.9
4	1140	22	98.1

C. Female Results

Using the same parameters as used for the males (Section IV-B), we performed the same recognition experiments on the 114 adult female speakers (57 training and 57 test) in TI-2. The results are listed in Table IV for section codebook rates 1 through 4. In general, the results for females were not as accurate as those for males. (For rate-4 section codebooks, average accuracy was 98.1 percent for the females and 98.8 percent for the males.) Also, it appears that increasing the section codebook rate for females stopped having a major effect on accuracy at a lower rate than was the case for males. Moreover, at each section codebook rate, the average classification distortions in these experiments were significantly smaller than the average distortions in the corresponding experiments with male speakers. These results all indicate that the space of LPC spectra has different characteristics for males and females. For the females, errors on *five, six,* and *nine* account for 14 of the 22 errors using rate-4 section codebooks.

D. Combined Male and Female Results

The separate results for males and females suggest that a rate-3 multisection codebook is adequate for recognition purposes. General differences in male and female vocal tract sizes, however, lead to characteristic formant shifts for the same speech sounds; thus, to attain the same recognition accuracy, mixed populations may require larger codebook sizes than single-sex populations. We examined this issue by performing a recognition experiment using the entire adult portion of TI-2. As before, we used two utterances of each digit from each speaker in the training portion to build codebooks, and we classified two utterances of each digit from each test speaker. The results for section codebook rates 1 through 6 are shown in Table V. Comparing these results with the earlier results (Tables III and IV) shows two things: the recognition accuracy has decreased (to 97.7 percent) and the required section codebook size has increased significantly.

TABLE V
COMBINED MALE AND FEMALE SPEAKER-INDEPENDENT RECOGNITION OF THE DIGITS: SECTION RATES 1 THROUGH 6

Section Rate	No. Class.	Errors	% Correct
1	2260	131	94.2
2	2260	94	95.8
3	2260	80	96.5
4	2260	69	96.9
5	2260	53	97.7
6	2260	58	97.4

TABLE VI
CONFUSION MATRIX FOR COMBINED MALE AND FEMALE DIGIT RECOGNITION: SECTION CODEBOOK RATE = 5

	0	1	2	3	4	5	6	7	8	9
0	225	.	.	.	.	.	.	.	.	1
1	.	219	.	.	3	1	.	.	.	3
2	2	.	220	.	.	.	2	2	.	.
3	.	.	.	226	.	.	.	.	.	.
4	.	3	.	.	223	.	.	.	.	.
5	.	1	.	.	2	218	.	.	.	5
6	.	.	2	1	.	.	215	5	3	.
7	.	.	.	.	.	.	2	220	1	3
8	.	.	2	1	.	.	6	.	217	.
9	.	2	.	.	.	.	.	.	.	224

Table VI is a confusion matrix for the rate-5 results. The most common errors were *five* → *nine, six* → *seven,* and *eight* → *six*. On examining the individual errors, we discovered that the endpoint detector had made a error on 37 of the 53 words that were misclassified. We hand marked the 37 words and reclassified them in the rate-5 section codebooks. Twenty-six of the errors were corrected including all the *six* errors, 3 of the *eight* errors, but only 1 of the *five* errors. Thus by eliminating the endpoint detection errors from the words incorrectly classified, about half of the total errors were eliminated and the average recognition accuracy rose to 98.8 percent.

Since correcting the endpoints on words that were incorrectly classified can only improve the average recognition accuracy, we also tried reclassifying words that contained endpoint errors but were correctly classified. We arbitrarily chose 66 such words (30 male utterances and 36 female utterances). These 66 words contain about 6 utterances of each digit. We corrected the endpoints, and using the same rate-5 section codebooks as above, we reclassified the 66 words. Only 1 error was introduced by hand marking the endpoints. This error was on the word five; it was classified as nine, which is a common error made by the multisection method (see Table VI.) Based on the above results, it is clear that accurate endpoint detection improves the performance of the multisection approach.

In these combined male and female experiments, rate-5 section codebooks were required to achieve accuracy that was close to the accuracy of the rate-3 male-only or rate-3 female-only results. We had expected a doubling of the required number of codewords per section, but not a quadrupling. As a check on the "quality" of the codewords designed from the combined male-and-female data, we classified the test data in codebooks made by merging the two rate-3 codebooks made from the single-sex training sequences. The resulting merged codebooks had 16 codewords per section, which is the same size as rate-4 section codebooks. The average recognition accuracy using the merged codebooks was 97.5 percent, 0.6 percent better than the results for the rate-4 section codebooks that were designed from the combined male and female training data (Table V).

These results suggest that merging the separately designed male and female codebooks provides a better representation of the combined training data than does a codebook designed from the combined male and female training sequence. This is not supported by the average distortion figures, however. We classified the male and female training data using each codebook set—the average classification distortion using the merged codebooks was 0.36; using the codebooks designed from the combined training data, the average distortion was 0.35. It appears that the two codebook types are roughly equivalent for representing the training sequence: the average distortions are about equal, and for a particular word, many of the codebook spectra in the two codebook types are almost identical. The merged codebooks, however, made 12 fewer errors than the standard codebooks (57 errors versus 69 errors), and we do not understand why the merged codebooks discriminated better than the standard codebooks in this test.

V. Speaker-Dependent Experiments

In this section, we describe the results of speaker-dependent experiments using TI-1. We began with a section codebook rate study of the multisection approach. We then compared two ways of using rate-0 multisection codebooks. Finally, we evaluated the effect of short training sequences. All the experiments described in this section used the 24-frame length normalized approach.

A. *Multisection Results*

In the speaker-independent study described in the previous section, good recognition performance required a section codebook rate of 4. It seems reasonable, however, that a smaller section codebook rate might suffice for speaker-dependent recognition. To evaluate this possibility, we performed speaker-dependent recognition experiments using the 16 speakers in TI-1. For each speaker, the first 10 utterances of each word were used as a training sequence. We used a compression factor of 4 and section codebook rates 0, 1, and 2.

Table VII contains the results for all 16 speakers. The first 8 are male and the last 8 are female, and the male results are slightly better than the female results. As one would expect, the average recognition accuracy improves with increases in section codebook rate. Using the two-sample Wilcoxon test to compare the rate-0 versus rate-1, rate-1 versus rate-2, and rate-0 versus rate-2 results, the significance levels for rejection of the null hypotheses of equal average recognition accuracies were 0.138, 0.133, and 0.031, respectively. Based on the Wilcoxon test results and the average recognition accuracies, we believe the use of rate-2 section codebooks significantly increases the recognition accuracy compared to rates 0 and 1.

TABLE VII
SECTION RATE STUDY FOR SPEAKER-DEPENDENT RECOGNITION

		Comp. Fact. = 4 Section Rate = 0	Comp. Fact. = 4 Section Rate = 1	Comp. Fact. = 4 Section Rate = 2
Speaker	No. Class.	% Correct	% Correct	% Correct
TBS	320	98.8	100.0	100.0
WMF	320	98.8	98.8	99.1
RLD	320	97.5	98.1	99.4
GRD	320	95.6	95.9	96.3
KAB	320	99.7	99.4	99.7
MSW	320	98.4	98.8	99.1
REH	320	97.8	98.8	99.1
RGL	320	100.0	100.0	100.0
CJP	320	95.9	97.8	97.8
DFG	320	95.3	97.5	99.1
ALK	320	99.4	99.4	99.7
HNJ	320	95.3	95.6	96.3
GNL	320	97.8	98.8	98.8
JWS	320	98.1	98.8	99.4
SJN	320	99.7	99.7	99.7
SAS	320	96.3	96.9	96.3
all	5120	97.8	98.4	98.7

The average recognition accuracy obtained with the rate-2 section codebooks was 98.7 percent. A confusion matrix for these results is shown in Table VIII. The most frequent errors were *go* ⟷ *no*, *stop* → *five*, and *start* → *five*. Most of the *go* and *no* classification errors were because of their spectral and temporal similarities. Many of the other classification errors can be attributed to time alignment problems caused by inadequacies of the endpoint detector.

To be more specific, we examined the errors made using the rate-2 section codebooks: there were 66 words incorrectly classified. The endpoints had been misidentified on 42 of those 6 words. We hand labeled the endpoints on those 42 words and reclassified them in the original codebooks. Thirty-eight of the 42 words were now correctly identified, and the average recognition accuracy increased to 99.5 percent. This improvement again points out the importance of accurate endpoint decisions.

In our previous single-section work [17], we performed a similar speaker-dependent classification experiment with TI-1. In that work, the 12 500 samples per second data was used together with the following analysis conditions: $N = 250$ points, analysis window = 250 points, analysis filter order = 16, preemphasis = 90 percent, and Hamming windowing. As in this study, we used the autocorrelation method of LPC and the first 10 utterances of each word for each speaker as training data. The recognition accuracy using single-section, rate-3 codebooks on the full bandwidth data was about the same as using multisection, rate-2 codebooks on the narrow bandwidth data: 98.8 and 98.7 percent, respectively. Based on reductions in both the analysis filter order and the section codebook rate, incorporating time-sequence information reduced the computational requirements by slightly more than a factor of 3, at the expense of doubling the memory required.

B. *Rate-0 Multisection Study*

The most remarkable aspect of the speaker-dependent results is the high recognition accuracy of the rate-0 code-

TABLE VIII
FULL DATABASE SPEAKER-DEPENDENT CONFUSION MATRIX: COMPRESSION FACTOR = 4, SECTION RATE = 2

	0	1	2	3	4	5	6	7	8	9	EN-TER	ER-ASE	GO	HELP	NO	RUB-OUT	RE-PEAT	STOP	START	YES
0	255	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	.	.	.
1	.	251	.	.	.	.	.	.	.	3	.	.	.	2	.	.	.	.	.	.
2	.	.	256	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
3	.	.	.	254	.	.	.	.	.	.	.	.	.	.	.	.	2	.	.	.
4	.	.	1	.	255	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	254	.	.	.	.	.	.	.	.	.	.	.	2	.	.
6	.	.	.	.	.	.	256	.	.	.	.	.	.	.	.	.	.	.	.	.
7	.	.	.	.	.	.	2	251	.	1	.	.	.	1	.	.	.	.	1	
8	.	.	.	1	.	.	.	.	252	.	1	1	.	.	.	.	1	.	.	.
9	.	.	1	.	.	.	.	.	.	255	.	.	.	.	.	.	.	.	.	.
ENTER	.	.	.	.	.	.	.	.	.	.	255	.	.	.	.	.	.	.	1	.
ERASE	.	.	.	.	.	.	.	.	.	1	.	254	.	.	.	.	1	.	.	.
GO	.	.	.	.	1	.	.	.	.	.	.	.	246	1	8	.	.	.	.	.
HELP	.	.	.	.	.	.	.	.	1	.	.	.	.	253	.	.	.	2	.	.
NO	.	2	.	.	.	.	.	.	.	.	.	.	5	.	248	1	.		.	
RUBOUT	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	256	.	.	.	
REPEAT	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	256	.	.	
STOP	.	.	.	.	.	8	.	1	.	.	.	.	.	2	.	.	.	245	.	.
START	.	.	.	2	.	5	2	.	.	.	.	1	.	.	.	.	.	.	246	.
YES	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	256

books (see Table VII). The multisection codebook for each word consists of only 6 codewords—one codeword per section—and the classification of an input utterance requires only one distortion computation per input frame for each vocabulary word. Moreover, the codebook generation consists simply of computing autocorrelations and averaging them, which is also easy to do quickly. Yet, despite these major simplifications, a recognition accuracy of 97.8 percent was achieved. Considering only the digits, the recognition accuracy was 99.5 percent.

Building references by linearly normalizing the training utterances to the same length, and then computing the average of a set of parameters for each frame in the normalized word, is an approach that many researchers evaluated before the introduction of dynamic programming and whole-utterance clustering techniques. Our rate-0, compression factor 4 (R0C4) approach is a modification of that normalize-the-utterance and average-each-frame (NUAF) approach using autocorrelations as the parameters. Because of the similarity between the two approaches, it is reasonable to ask if our R0C4 approach is any better than the old NUAF approach.

In the terminology of this paper, the NUAF approach corresponds to using rate-0, compression factor 1 (R0C1) codebooks. So, we designed R0C1 codebooks and evaluated them. Based on the speaker-independent parameter study results, we expected the larger compression factor codebooks (R0C4) to perform better than the smaller compression factor codebooks (R0C1).

Table IX contains the R0C1 results along with the previous R0C4 results from Table VII. Each compression factor 4 result is better than or equal to the compression factor 1 result except for speaker WMF, and using the Wilcoxon test on the two samples, the significance level for rejection of the null hypothesis of equal average recognition accuracies is 0.159. We believe the improved performance using a compresssion factor of 4 is because of two things: the slowly varying nature of speech spectra and the freedom from strict time alignment that a compression factor of 4 allows. Apparently, averaging the spectra in the training sequence over small sections of a word produce reference spectra that characterize a speaker's variation in pronunciation better than averaging over a single frame. Although the significance level for rejection of the null hypothesis is somewhat large, the amount of storage for each codebook is reduced and the recognition accuracies are better using a compression factor of 4.

TABLE IX
COMPRESSION FACTOR STUDY FOR SPEAKER-DEPENDENT RECOGNITION: SECTION RATE = 0

		Comp. Fact. = 1 Section Rate = 0	Comp. Fact. = 4 Section Rate = 0
Speaker	No. Class.	% Correct	% Correct
TBS	320	97.8	98.8
WMF	320	99.1	98.8
RLD	320	96.9	97.5
GRD	320	94.7	95.6
KAB	320	99.1	99.7
MSW	320	98.4	98.4
REH	320	97.8	97.8
RGL	320	99.4	100.0
CJP	320	95.6	95.9
DFG	320	95.3	95.3
ALK	320	98.1	99.4
HNJ	320	94.1	95.3
GNL	320	96.9	97.8
JWS	320	97.2	98.1
SJN	320	98.8	99.7
SAS	320	95.9	96.3
all	5120	97.2	97.8

C. Short Training Sequences

Many speaker-dependent isolated word recognition devices on the market today use from 1 to 3 training utterances to train the system [32]. Although our previous results [17] suggested the inadequacy of short training sequences, we confirmed this expectation. Using a

TABLE X
SPEAKER-DEPENDENT TRAINING SEQUENCE STUDY: COMPRESSION FACTOR = 4

		1 Utterance Training Seq. (Unclustered)	2 Utterance Training Seq. (Clustered)	10 Utterance Training Seq. (Clustered)
Speaker	No. Class.	% Correct	% Correct	% Correct
TBS	320	95.0	95.9	100.0
WMF	320	89.7	97.8	99.1
RLD	320	88.8	92.8	99.4
CJP	320	90.3	93.4	97.8
all	1080	90.9	95.0	99.1

compression factor of 4 and the first 2 utterances of each word as the training sequence, we classified the same 320 utterances as above for each of the 16 speakers in TI-1. The average recognition accuracies were 94.6, 95.6, and 95.7 percent for rate-0, rate-1, and rate-2 multisection codebooks, respectively. This is a decrease of about 3 percent at each rate relative to the results using 10-utterance training sequences (see Table VII).

Finally, we performed a recognition experiment on 4 speakers using 1 utterance training sequences. We used unclustered codebooks to retain all the information in the training data, and we used a compression factor of 4. These results along with the 2- and 10-utterance training sequence, rate-2 results are shown in Table X. The effect of using only one training utterance is dramatic. The average recognition accuracy for this 4-speaker subset has fallen to 90.9 percent.

These results using short training sequences simply emphasize what is commonly known: there is much variability in a speaker's pronunciation of a particular word. Hybrid approaches, however, can alleviate the problems inherent in short training sequences. For example, we recently demonstrated a method that augments a standard set of speaker-independent codebooks with codewords derived from short, speaker-specific training sequences [39].

VI. COMPUTATIONAL AND MEMORY CONSIDERATIONS

It is interesting to compare the computational and memory requirements of the multisection VQ approach to those of DTW for the classification of an unknown input utterance. As we pointed out earlier, the requirements for the DTW approach can be substantially reduced by incorporating VQ into the DTW procedure, albeit with reduced recognition accuracy [16], [19], but we do not consider that case here. Our intention is to compare the computational and memory requirements of the multisection VQ with that of "classical" DTW [33]. Savings obtained by tracking the average distortion during classification to reject several of the hypotheses or using table storage and lookup are also not considered.

In this analysis, we consider only the length-normalized approach. Let M be the LPC analysis filter order, N_{SC} be the number of codewords per section codebook, n be the compresssion factor, and L_N be the normalization length. Then the memory required for a multisection codebook is

$$N_{SC} \operatorname{ceil}\left[\frac{L_N}{n}\right](M + 1)$$

real numbers, where ceil $[X]$ is the smallest integer greater than or equal to X. Since the input word is normalized to L_N frames, classification requires $N_{SC} L_N$ distortion computations per multisection codebook.

In DTW approaches, the reference template and the input utterance are often linearly normalized to the same length L before doing DTW [33]. High recognition accuracies can then be achieved with αL^2 distortion computations per reference template, where α is in the range 0.20 to 0.35 [33]. Each reference template requires L storage locations, and to achieve high recognition accuracies, several reference templates per vocabulary word are normally stored. For speaker-dependent recognition, the number of reference templates Q is usually one or two; for speaker-independent recognition, Q is normally about ten [34].

It follows that the ratio D of the number of distortion calculations required by the VQ approach to the number required by the DTW approach is about $D \approx N_{SC}L_N/\alpha L^2 Q$. For fixed size codebooks with $N_{SC} = 2^{R_{SC}}$, where R_{SC} is the section codebook rate, and for a nominal value of $\alpha \approx 0.25$, the ratio becomes

$$D \approx \frac{2^{R_{SC}-3}}{Q}. \tag{6}$$

For our best speaker-dependent results on the 20-word vocabulary — 98.7 percent correct using a section codebook rate $R_{SC} = 2$ — (6) shows the ratio of distortion computations to be $1/2Q$. Since Q is usually 1 or 2 for the speaker-dependent case, this shows that the multisection VQ approach requires fewer distortion computations than DTW. The 98.7 percent speaker-dependent recognition accuracy of the multisection approach is comparable with that achieved by other approaches on this database [26].

For speaker-independent recognition of the digits, the multisection approach required the rate $R_{SC} = 4$. For this case, (6) shows the ratio of distortion computations to be $2/Q$. Since Q is normally about 10 for the speaker-independent case, this shows that the multisection approach requires 80 percent fewer distortion computations than DTW.

The ratio W of memory locations required by the multisection approach to the number required by the DTW approach is

$$W \approx \frac{N_{SC} \operatorname{ceil}\left[\frac{L_N}{n}\right]}{L_N Q},$$

where the length of a DTW reference L has been assumed equal to the normalization length L_N. Using a $L_N = 32$, an $n = 0.2L_N$ and substituting $2^{R_{SC}}$ for N_{SC},

$$W \approx \frac{2^{R_{SC}-2.7}}{Q}. \tag{7}$$

Equation (7) shows that for reasonable values of Q and R_{SC}, speaker-dependent and speaker-independent recognition using the multisection approach requires about one-half to one-fourth the memory that DTW requires.

The software for these experiments was written in Fortran-77 and run on a DEC VAX11/750 with a floating point accelerator. Starting with the autocorrelations from a 224-utterance training sequence, generating the fixed-size, rate-4, multisection codebooks required about 14.5 min of execution time each. Classification of a single utterance with these codebooks took about 0.2 s per codebook—about five times faster than our previous approach to speaker-independent recognition [17]. The speedup is the result of a combination of factors: the section codebooks are smaller than the previous single-section codebooks (16 codewords instead of 32), the narrower bandwidth data (4000 Hz vs 6250 Hz) allowed a reduction in the LPC filter order from 16th to 10th, and autocorrelations were computed over a 16 ms window instead of a 20 ms window. Since all the software was designed for research purposes, specially designed programs should run considerably faster.

VII. Summary and Discussion

In comparison to our previous single-section results [17], the incorporation of time-sequence information into the VQ recognition procedure has improved recognition performance. For male speaker-independent recognition, the average recognition accuracy for the 20-word vocabulary increased from 88 to 97 percent with a factor of 4 reduction in computational complexity. For combined male-and-female speaker-independent recognition of the digits, accuracy was 97.7 percent. For speaker-dependent recognition, the accuracies of the multi- and single-section approachs were about the same, but the multisection approach required only half the number of distortion computations. The costs for the computational and accuracy improvements of the multisection approach are a slightly more complicated control structure and an increase in memory for codebook storage.

Perhaps the most remarkable multisection VQ result was the 97.8 percent (99.5 percent for digits) speaker-dependent recognition accuracy for the rate-0 section codebooks. Only six spectra are used to characterize each vocabulary word, classification requires only one distortion computation per input speech frame per vocabulary word, and the codebook design requires no clustering.

The memory requirements and computational complexity of the speaker-dependent, multisection approach are about $\frac{1}{2}$ to $\frac{1}{4}$ those of the classical DTW approach. For speaker-independent recognition of the digits using rate-4 section codebooks, the multisection approach requires only about $\frac{1}{4}$ the memory and $\frac{1}{5}$ the distortion computations of DTW.

As general conclusions about the multisection VQ approach, we offer the following:

1) all utterances should be length normalized before processing;

2) the normalization length should be as long as computational constraints permit (up to the maximum word length expected);

3) the analysis conditions should provide frame overlap;

4) for speaker-independent recognition of males only or females only, a section codebook rate of 3 performs well; for combined males and females, a higher rate is needed;

5) for speaker-dependent recognition, a section codebook rate of 2 is required;

6) short training sequences cannot be used;

7) accurate endpoint detection is important.

There are several ways in which the multisection VQ method might be applied in building speech recognition systems. First, the method could be used directly for small vocabularies to implement inexpensive recognizers that are quite accurate. For larger vocabularies, it might be feasible to implement a hierarchical approach. Such an approach could begin by using an initial set of VQ codebooks to classify an input word as belonging to a particular subset of the recognition vocabulary. Each of the initial codebooks would be designed to encode, with low distortion, a group of phonetically similar words. Additional sets of codebooks would then be used to refine the classification down to an individual word in the vocabulary. Another possibility for larger vocabularies, as well as for systems that require extremely high recognition accuracy, is to use the multisection VQ classifier as a preprocessor to filter out unlikely candidates in a more elaborate system that uses additional methods. Single-section VQ codebooks have already been used in this way [35], and it is likely that multisection codebooks would provide a faster and more accurate preprocessor.

The success of the multisection approach is due primarily to two things. First, VQ codebooks are an efficient representation of the training data. Second, multisection codebooks allow flexibility in the time alignment of an input utterance with a codebook, but they enforce sectional time alignment. In fact, there is an analogy in the time alignment procedures of DTW and multisection VQ. Neither enforces a strict sequential frame by frame comparison of the input and references, and both find locally a best path through the reference. The analogy quickly breaks down, but it is clear that the nonlinear time alignment allowed by both approaches contributes to their success.

Our original single-section VQ approach tried to model each vocabulary word as a discrete memoryless source [17]. Although the results were good, this model is, of course, naive. A better source model for an isolated word is a Markov model, and many researchers have used this idea [36], [37], [16], [19]. Multisection VQ is an ad hoc way of incorporating memory. It can be viewed as a one-step Markov model with transition probabilities that are either zero or one for moving to the next state or section. It would be more satisfying, and we suspect more accurate, if the states and the state representations for a word were determined by the same criterion as that used in designing a memoryless VQ codebook—minimizing the distortion between the training data and the representation. Some steps in this direction have been made.

Ostendorf and Gray have developed an algorithm for designing both a separate zero memory quantizer for each of a finite number of states and a set of next-state functions depending only on the current state and codeword to update the state [38]. Using this algorithm, a separate finite-state vector quantizer could be designed for each vocabulary word, and an unknown input utterance could be classified by encoding it in each of the finite-state vector quantization codebooks, just as is now done with the multisection codebooks. Since time-sequence information is implicit in the next-state function, and since a state codebook is likely to be smaller than a section codebook, the recognition accuracy should improve and the computational complexity should decrease.

Acknowledgment

We thank A. Buzo and R. Johnson for helpful discussions, R. M. Gray for providing some of the vector quantization software, T. Schalk and G. Leonard for their help in obtaining the databases, and the referees for helpful comments and suggestions.

References

[1] T. Berger, *Rate Distortion Theory: A Mathematical Basis for Data Compression.* Englewood Cliffs, NJ: Prentice-Hall, 1971.

[2] R. M. Gray, "Vector quantization," *ASSP Magazine*, pp. 4–29, Apr. 1984.

[3] A. Buzo, A. H. Gray, Jr., R. M. Gray, and J. D. Markel, "Speech coding based upon vector quantization," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-28, pp. 562–574, Oct. 1980.

[4] R. M. Gray, A. H. Gray, Jr., G. Rebolledo, and J. E. Shore, "Rate-distortion speech coding with a minimum discrimination information distortion measure," *IEEE Trans. Inform. Theory,* vol. IT-27, pp. 708–721, Nov. 1981.

[5] G. Rebolledo, R. M. Gray, and J. P. Burg, "A multirate voice digitizer based upon vector quantization," *IEEE Trans. Commun.*, vol. COM-30, pp. 721–727 Apr. 1982.

[6] A. Gersho and B. Ramamurthi, "Image coding using vector quantization," *Proc. ICASSP'82 IEEE Int. Conf. Acoust., Speech, Signal Processing,* Paris, France, May 1982, IEEE 82CH1746-7, pp. 428–431.

[7] R. L. Baker and R. M. Gray, "Image compression using nonadaptive spatial vector quantization," in *Conf. Rec. Sixteenth Asilomar Conf. Circuits, Syst. Comput.*, Oct. 1982, pp. 55–61.

[8] R. Hamabe, Y. Yamada, M. Murata, and T. Namekawa, "A speech recognition system using inverse filter matching technique" (in Japanese), in *Proc. Annu. Conf. Inst. Television Engineers*, Kyushu University, Japan, June 1981.

[9] J. E. Shore and D. Burton, "Discrete utterance speech recognition without time normalization," in *Proc. ICASSP'82 IEEE Int. Conf. Acoust., Speech, Signal Processing,* Paris, France, May 1982, IEEE 82CH1746-7, pp. 907–910.

[10] ——, "Discrete utterance speech recognition without time normalization—Recent results," in *Proc. 1982 6th Int. Conf. Pattern Recognition,* IEEE 82CH1801-0, Oct. 1982, pp. 582–584.

[11] A. Buzo, C. Riviera, and H. Martinez, "Discrete utterance recognition based upon source coding techniques," in *Proc. ICASSP'82 IEEE Int. Conf. Acoust., Speech, Signal Processing,* Paris, France, May 1982, IEEE 82CH1746-7, pp. 539–542.

[12] R. Billi, "Vector quantization and Markov source models applied to speech recognition," in *Proc. ICASSP'82 IEEE Int. Conf. Acoust., Speech, Signal Processing,* Paris, France, May 1982, pp. 574–577.

[13] D. K. Burton, J. E. Shore, and T. J. Buck, "A generalization of isolated word recognition using vector quantization," in *Proc. ICASSP'83 IEEE Int. Conf. Acoust., Speech, Signal Processing,* Boston, MA, Apr. 1983, IEEE 83CH1841-6, pp. 1021–1024.

[14] N. Sugamura, K. Shikano, and S. Furiu, "Isolated word recognition using phoneme-like templates," in *Proc. ICASSP'83 IEEE Int. Conf. Acoust., Speech, Signal Processing,* Boston, MA, Apr. 1983, pp. 723–726.

[15] R. Pieraccini and R. Billi, "Experimental comparison among data compression techniques in isolated word recognition," in *Proc. ICASSP'83 IEEE Int. Conf. Acoust., Speech, Signal Processing,* Boston, MA, Apr. 1983, pp. 1025–1023.

[16] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *Bell Syst. Tech. J.*, vol 62, no. 4, pp. 1075–1105, Apr. 1983.

[17] J. E. Shore and D. K. Burton, "Discrete utterance speech recognition without time alignment," *IEEE Trans. Inform. Theory,* vol. IT-29, pp. 473–491, July 1983.

[18] D. K. Burton, J. T. Buck, and J. E. Shore, "Parameter selection for isolated word recognition using vector quantization," in *Proc. ICASSP'84 IEEE Int. Conf. Acoust., Speech, Signal Processing,* San Diego, CA, Mar. 1984, IEEE 84CH1945-5.

[19] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, " On the use of hidden Markov models for speaker-independent recognition of isolated words from a medium-size vocabulary," *AT&T Bell Lab. Tech. J.*, vol. 63, no. 4, Apr. 1984.

[20] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84–95, Jan. 1980.

[21] C. E. Shannon, "Coding theorems for a discrete source with a fidelity criterion," in *Information and Decision Processes,* R. E. Machol, Ed. New York: McGraw-Hill, 1960, pp. 93–126.

[22] B.-H. Juang, D. Y. Wong, and A. H. Gray, Jr., "Distortion performance of vector quantization for LPC voice coding," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-30, pp. 294–303, Apr. 1982.

[23] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-23, pp. 67–72, Feb. 1975.

[24] R. M. Gray, A. Buzo, A. H. Gray, Jr., and Y. Matsuyama, "Distortion measures for speech processing," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-28, pp. 367–376, Aug. 1980.

[25] T. E. Tremain, "The government standard linear predictive coding alogrithm: LPC-10," *Speech Technology,* vol. 1, pp. 40–49, Apr. 1982.

[26] G. R. Doddington and T. B. Schalk, "Speech recognition: Turning theory to practice, *IEEE Spectrum,* vol. 18, no. 9, pp. 26–32, Sept. 1981.

[27] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proc. ICASSP'84 Conf.*, Mar. 1984, pp. 42.11.1–42.11.4.

[28] L. R. Rabiner and M. Sambur, "An algorithm for determining the endpoints of isolated utterances," *Bell Syst. Tech. J.*, vol. 54, pp. 297–315, Feb. 1975.

[29] L. Lamel *et al.*, "An improved endpoint detector for isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-29, pp. 777–785, Aug. 1981.

[30] R. V. Hogg and E. A. Tanis, *Probability and Statistical Inference.* New York: Macmillan, 1977.

[31] B. A. Dautrich, L. R. Rabiner, and T. B. Martin, "The effects of selected signal processing techniques on the performance of a filter-bank-based isolated word recognizer," *Bell Syst. Tech. J.*, vol. 62, no. 5, May–June 1983.

[32] W. A. Lea, "Selecting the best speech recognizer for the job," *Speech Technology,* vol. 1, pp. 10–22, 27–29, Jan./Feb. 1983.

[33] C. S. Meyers, L. R. Rabiner, and A. E. Rosenberg, "Performance tradeoffs in dynamic time warping algorithms for isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-28, pp. 623–635, Dec. 1980.

[34] L. R. Rabiner and J. G. Wilpon, "Speaker-independent isolated word recognition for a moderate size (54 word) vocabulary," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-27, p. 583–587, Dec. 1979.

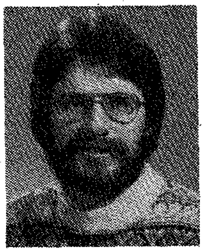
[35] K.-C. Pan, "Isolated word recognition based upon vector quantization techniques," Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, 1984.

[36] J. K. Baker, "The DRAGON system—An overview," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-23, pp. 24–29, Feb. 1975.

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

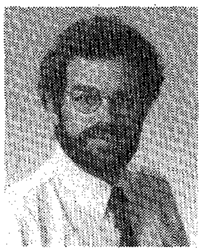
[38] M. Ostendorf and R. M. Gray, "An algorithm for the design of labeled-transition finite-state vector quantizers," *IEEE Trans. Commun.*, to be published.

[39] D. K. Burton and J. E. Shore, "Speaker-dependent isolated word recognition using speaker-independent vector quantization codebooks augmented with speaker-specific data," *IEEE Trans. Acoust., Speech, Signal Processing,* vol. ASSP-33, pp. 440–443, Apr. 1985.



David K. Burton (M'78) was born in Washington, DC, on September 8, 1952. He received the B.S. and M.S. degrees in electrical engineering from the University of Maryland, College Park, MD, in 1974 and 1981, respectively.

Previously, he worked at Presearch, Inc., Crystal City, VA, on modeling the performance of antennas in cluttered environments, and at Amecom, College Park, MD, on frequency agile exciters. In 1980 he joined the Naval Research Laboratory, Washington, DC, and is currently applying information-theoretic methods to signal processing, speech processing, spectrum analysis, and array processing.



John E. Shore (M'72–SM'81) was born in England on September 2, 1946. He received the B.S. degree in physics from Yale University, New Haven, CT, in 1968, and the Ph.D. degree in theoretical physics (statistical mechanics) from the University of Maryland, College Park, in 1974.

In 1968 he joined the Naval Research Laboratory, Washington, DC, where he is currently a Research Scientist in the Computer Science and Systems Branch, Information Technology Division. He is also a Senior Scientist at Entropic Processing, Inc., Cupertino, CA. His previous research interests include computer architecture, dynamic memory allocation, programming language design, software engineering, and text-to-speech translation. His current interests include information theory, queuing theory, search theory, system modeling, pattern recognition, spectrum analysis, and speech processing.

Joseph T. Buck (S'77–M'79) was born in Ogdensburg, NY, on December 19, 1957. He received the B.E.E. degree from Catholic University of America, Washington, DC, in 1978, and the M.S. degree in computer science from George Washington University, Washington, DC, in 1982.

From 1979 to 1982 he was employed by ORI, Inc., Silver Spring, MD. From 1982 to 1984 he was with the Naval Research Laboratory, where he participated in the research described in this paper. He is currently Manager of Software for the Speech Division of Entropic Processing, Inc., Cupertino, CA. He also participates in speech processing research.