Text-Dependent Speaker Verification Using Vector Quantization Source Coding

DAVID K. BURTON, MEMBER, IEEE

Abstract—**Several vector quantization approaches to the problem of text-dependent speaker verification are described. In each of these approaches, a source codebook is designed to represent a particular speaker saying a particular utterance. Later, this same utterance is spoken by a speaker to be verified and is encoded in the source codebook representing the speaker whose identity was claimed. The speaker is accepted if the verification utterance's quantization distortion is less than a prespecified speaker-specific threshold. The best approach achieved a 0.7 percent false acceptance rate and a 0.6 percent false rejection rate on a speaker population comprising 16 admissible speakers and 111 casual imposters. The approaches are described, and detailed experimental results are presented and discussed.**

I. Introduction

SPEAKER verification by machine consists of automatically authenticating the identity claimed by a speaker, given only samples of the speaker's voice. It has been an area of active study for more than twenty years resulting in two categories of approaches. In one, verification decisions are based on speech that is selected by the speaker and not known ahead of time by the verification system; this is called *text-independent* verification. In the other category, the verification system is trained on *a particular utterance*. This *same utterance* is later spoken by the individual in question and a verification decision is made based on this utterance; this is called *text-dependent* speaker verification. In this paper, we describe and evaluate several new approaches to the *text-dependent* speaker verification problem.

A typical approach to text-dependent speaker verification consists of selecting parameters that can be derived from the speech waveform and then representing each speaker by a time-series of these parameters (called a reference template) obtained from a particular utterance. The parameters are normally chosen with the hope that they reflect speaker-specific, organic differences in the structure of the vocal apparatus or, perhaps, that the time series of parameters will reflect learned differences in the use of the vocal apparatus to produce a particular utterance. After obtaining a reference for each speaker to be verified, an unknown speaker claims an identity and speaks the appropriate utterance. This utterance is analyzed, and a time-series of parameters is obtained. The unknown speaker's parameters are then aligned in time with the reference stored for the speaker whose identity was claimed and the decision to accept or reject the speaker is based on a measure of the similarity between the two time-series of parameters. Examples of parameters that have been used in this way are pitch [1], short-time energy [1], short-time spectra [2], and linear predictive coding (LPC) coefficients or parameters derived from these coefficients [3].

In addition to the template matching approach described above, statistical methods are sometimes used [4], [5]. These methods use large amounts of training data to estimate the underlying probability densities of parameters that are chosen to represent a speaker. Once the probability densities are specified, statistical decision theory methods [6] are used to verify a speaker.

We approach the text-dependent speaker verification problem from a different viewpoint. We consider a speaker of a particular utterance as an information source, and we model this information source by using a standard information-theoretic source coding method called *vector quantization* (VQ). VQ is a source coding technique [7] that has been used successfully in both speech coding [8] and speech recognition [9]–[11]. In VQ, each source vector is coded as one of a prestored *set of codewords*, called a *codebook*, by finding the codeword that minimizes the distortion between itself and the source vector. For speech, a VQ codebook is designed from a training sequence containing typical speech [12]. The training sequence is divided into frames (typically 20 ms), linear predictive analysis is done on each frame, and a clustering algorithm is used on this set of LPC coefficient vectors to obtain a codebook of representative vectors or codewords. The codebook is designed to minimize the average quantization distortion between itself and the training sequence.

To use VQ source coding in speaker verification, we represent each speaker by a VQ codebook designed from a training sequence composed of repetitions of a particular utterance. Later, this same utterance is spoken by an unknown speaker with a claimed identity. This test utterance is coded in the codebook representing the speaker whose identity was claimed, and the resulting quantiza-

Manuscript received May 16, 1985; revised June 24, 1986. This work was done while the author was employed at the Naval Research Laboratory, Washington, DC.

The author is with Entropic Speech Incorporated, Washington Research Laboratory, Washington, DC 20003, a division of Entropic Processing, Inc., Cupertino, CA 95014.

IEEE Log Number 8610885.

tion distortion is compared to a threshold. If the distortion is less than the threshold, the speaker is accepted.

In addition to our source coding point of view, our speaker verification approach is quite different from other approaches in several ways. No attempt is made to align-in-time a test sequence with a stored reference sequence (indeed, no reference sequence exists), and no explicit estimate is made of the underlying probability density function of the parameters chosen to represent the speaker. Our verification procedures are, however, closely related to optimal information-theoretic methods of classification that use the *information dissimilarity* between two vectors as a discrimination measure [13], [14]. Preliminary results were reported in [15].

We previously used VQ source coding in isolated word recognition [9], [16], [17]. The methods used in those approaches to represent a word (design a codebook) and to compare an unknown input word to the stored codebooks (classify an input utterance) are the same as the ones described in this paper to represent and verify a speaker. The differences are in the application of the ideas and in the use of thresholds to make decisions.

In related work, Li and Wrench represented speakers with vectors from a general-speech VQ codebook in a text-independent speaker recognition scheme [18], and much more recently and akin in spirit to our approaches, Soong *et al.* [19] and Shikano [20] reported on text-independent speaker recognition approaches. In Soong's approach, each speaker is represented by a VQ codebook designed to represent that speaker saying the names of the 10 decimal digits. An unknown speaker then says any of the 10 digits, and the average quantization distortion resulting from encoding the spoken digits is used as a discrimination measure to identify the speaker. In Shikano's approach, each speaker is represented by a VQ codebook designed from unconstrained continuous speech, and different speech is used for recognition. Reported results for both methods are quite good.

The rest of this paper is organized as follows. Section II describes three ways to represent a speaker by using VQ source coding. Section III explains our speaker verification approach. Section IV presents experimental results, and Section V concludes with a summary and general discussion.

II. Background

In this section we briefly describe three ways to design a source model of a speaker; for detailed descriptions of the methods, see [9], [16], and [17]. Following these descriptions is a list of the distortion measures and LPC parameters used in the speaker verification experiments.

First we establish some notation and define some terms. Upper and lower case Roman and italic letters (e.g., n, N, q, Q) denote scalars; lower case italic letters with bars (e.g., $\bar{c}$) denote vectors; upper case italic letters with bars denote sets of vectors (e.g., $\overline{C} = \{\bar{c}_1, \bar{c}_2, \cdots, \bar{c}_N\}$); bold lower case Roman letters denote sequences of vectors (e.g., $\mathbf{c} = [\bar{c}_i; i = 1, \cdots, K]$); and bold upper case Roman letters denote sets of vector sequences (e.g., $\mathbf{C} = \{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$). In what follows, a set means an unordered collection of elements; a sequence means a time-ordered collection of elements.

Throughout, all vectors consist of LPC coefficients and a gain term. $\mathbf{t} = [\bar{t}_1, \bar{t}_2, \cdots, \bar{t}_P]$ is a P-vector training sequence obtained from M catenated repetitions of an utterance by a speaker. $\mathbf{v} = [\bar{v}_1, \bar{v}_2, \cdots, \bar{v}_L]$ is an L-vector test sequence corresponding to an utterance obtained from a speaker for verification purposes. $\overline{C}$ represents a VQ codebook, and finally, $\mathbf{C}$ represents a matrix quantization codebook.

A. Single Section Vector Quantization

For speaker verification, a single-section VQ codebook $\overline{C}$ is designed to minimize the average distortion that results from encoding a training sequence $\mathbf{t}$

$$\sum_{p=1}^{P} d(\bar{t}_p, \bar{c}_B), \tag{1}$$

where $\bar{c}_B$ is the codeword that results from encoding speech segment $\bar{t}_p$,

$$d(\bar{t}_p, \bar{c}_B) = \min_i d(\bar{t}_p, \bar{c}_i),$$

and d is an appropriate vector distortion measure. This codebook represents a speaker saying a particular word.

The average quantization distortion D_{avg} that results from coding a verification utterance $\mathbf{v}$ in codebook $\overline{C}$ is

$$D_{avg} = \frac{1}{L} \sum_{l=1}^{L} d(\bar{v}_l, \bar{c}_B). \tag{2}$$

We use this average quantization distortion in making the verification decision.

This approach is called *single section* to distinguish it from the approach described in the next section, in which each speaker is represented by a sequence of single section codebooks.

B. Multisection Vector Quantization

In multisection VQ, we represent each speaker by a time-ordered sequence of single section codebooks, which we call a multisection codebook. A speaker is verified by dividing a verification utterance $\mathbf{v}$ into sections corresponding to the sections of the multisection codebooks, doing VQ section by section with the appropriate multisection codebook, and computing the average distortion.

To be more specific, let F_q be the number of frames in the qth utterance in the training sequence $\mathbf{t}$ for multisection codebook $\overline{C}$, where $q = 1, \cdots, M$; and let U_{fq} be the fth frame in the qth training utterance, where $f = 1, \cdots, F_q$ and $P = \Sigma_{q=1}^{M} F_q$ is the total number of training vectors in $\mathbf{t}$. The multisection codebook $\overline{C}$ consists of a sequence of VQ *section codebooks* $\overline{C}_j$, where the section codebook $\overline{C}_j$ is designed using (1) and n frames from each training utterance. That is, $\overline{C}_j$ is designed from the frames U_{fq}, where $f = (j-1)n + 1, \cdots, jn$, and $q = 1, \cdots, M$. For example, $\overline{C}_1$ is designed from the first n frames of

each training utterance, $\bar{C}_2$ from the second n frames, and so on. We call n the *section length*—it is the number of frames that are spanned per section. Finally, let $\{\bar{c}_{j1}, \bar{c}_{j2}, \cdots, \bar{c}_{jN_j}\}$ be codewords in section codebook $\bar{C}_j$.

D_{avg} is the average distortion that results from coding the verification utterance $\mathbf{v}$ with the codebook $\bar{C}$,

$$D_{avg} = \frac{1}{L} \sum_{j=1}^{S} d_j, \tag{3}$$

where S is the number of section codebooks in $\bar{C}$,

$$d_j = \sum_{l=(j-1)n+1}^{\min[jn, L]} \min_i d(\bar{v}_l, \bar{c}_{ji})$$

is the total distortion from coding the jth section of the utterance $\mathbf{v}$ with the jth section codebook $\bar{C}_j$ of $\bar{C}$, L is the number of vectors in $\mathbf{v}$, and n is the section length. The verification decision is made using this average distortion.

C. Matrix Quantization

In matrix quantization, instead of coding a *single* source vector in a codebook containing characteristic vectors, we code a *time-ordered sequence* of source vectors in a codebook containing characteristic vector sequences. Given $\mathbf{t}$, we find the matrix quantization codebook $\mathbf{C}$ containing codework matrices (or vector sequences) $\mathbf{c}_j = [\bar{c}_{j1}, \bar{c}_{j2}, \cdots, \bar{c}_{jK}]$ that minimizes

$$\sum_{p=1}^{P-K+1} D(\mathbf{t}_p, \mathbf{c}_B),$$

where $\mathbf{c}_B$ is the codeword matrix that results from coding the sequence of training vectors

$$\mathbf{t}_p = [\bar{t}_p, \bar{t}_{p+1}, \cdots, \bar{t}_{p+K-1}],$$

by using the nearest neighbor rule

$$D(\mathbf{t}, \mathbf{c}_B) = \min_j D(\mathbf{t}, \mathbf{c}_j),$$

and where the distortion between a speech segment $\mathbf{t}$ and the jth codeword matrix is

$$D(\mathbf{t}, \mathbf{c}_j) = \sum_{l=1}^{K} d(\bar{t}_l, \bar{c}_{jl}). \tag{4}$$

We call K the *codeword matrix size*. The MQ codebook design algorithm [21] we used is a generalized version of the VQ design algorithm developed by Linde *et al.* [12].

To use MQ in speaker verification, we represent each speaker by a codebook $\mathbf{C}$, just as in the VQ approaches above. A verification utterance is processed by dividing it into overlapping sequences of K frames, coding each K frame sequence in the speaker-codebook $\mathbf{C}$, and computing the average quantization distortion between the utterance and the codebook. To be specific, for a verification utterance $\mathbf{v}$, the average distortion resulting from coding it with codebook $\mathbf{C}$ is

$$D_{avg} = \frac{1}{L - K + 1} \sum_{l=1}^{L-K+1} D(\mathbf{v}_l, \mathbf{c}_B) \tag{5}$$

where $\mathbf{v}_l = [\bar{v}_l, \bar{v}_{l+1}, \cdots, \bar{v}_{l+K-1}]$.

D. Distortion Measures

Based on results from previous work on isolated word recognition [9], we used the *gain normalized Itakura–Saito* distortion measure (d_{GN}) in (1) and (4) to generate codebooks. For power spectrum estimates f and $\hat{f}$ having the autoregressive (LPC) form

$$f(\theta) = \frac{\sigma^2}{|A(z)|^2},$$

where

$$A(z) = \sum_{k=0}^{M} a_k z^{-k}$$

and $z = \exp(i\theta)$, the d_{GN} distortion is given by

$$d_{GN}(f, \hat{f}) = \frac{\alpha}{\sigma^2} - 1,$$

where

$$\alpha = r(0)\, \hat{r}_a(0) + 2 \sum_{n=1}^{M} r(n)\, \hat{r}_a(n),$$

$$\hat{r}_a(n) = \sum_{i=0}^{M-n} \hat{a}_i \hat{a}_{i+n},$$

and where $r(n)$ are the time-domain autocorrelations of $f(\theta)$.

We chose the *gain optimized Itakura–Saito* distortion measure (d_{GO})

$$d_{GO}(f, \hat{f}) = \ln(\alpha) - \ln(\sigma^2),$$

for the verification distortion measure in (2), (3), and (5), again based on results from previous work [9]. d_{GN} would also be a good choice [9]. Properties of these distortion measures are discussed in [22].

E. LPC Parameters

LPC parameters for both codebook generation and speaker verification were generated using the autocorrelation method of linear predictive analysis with Hamming windowing. We chose analysis conditions for compatibility with the Navy's 2.4-kbits/s LPC-10 system [23]: analysis window width = 128 points, filter order = 10, and preemphasis = 94 percent.

III. Speaker Verification Approach

Usually, no information is available for the characteristics of specific unacceptable speakers, and the main problem in applying these source coding approaches to speaker verification is in formulating a criterion for rejecting a speaker. To decide whether to reject a speaker (given an utterance), we associate a threshold with each speaker codebook. An unknown speaker (utterance) is re-

jected if its distortion exceeds the threshold. To design thresholds for a speaker, we estimate parameters for two Gaussian distributions: the *in-class* distribution of distortions obtained by encoding utterances from a speaker in his or her codebook, and the *out-of-class* distribution of distortions resulting from encoding utterances spoken by other speakers. We chose the threshold to equalize the overlap area of the two distributions, thus equalizing the expected numbers of imposter acceptances (false acceptances) and rejections of admissible speakers (false rejections).

In more detail, the threshold computation is as follows. For each speaker, encode that speaker's training data with his or her codebook. Compute the mean distortion μ_i^{in} resulting from encoding the training data from speaker i in speaker i's codebook, and compute the corresponding standard deviation σ_i^{in}. Also compute μ_i^{out}, the mean distortion resulting from encoding utterances *not* spoken by speaker i using the codebook for speaker i, and the corresponding standard deviation σ_i^{out}. To equalize the number of false acceptances and false rejections, the threshold T_i is chosen to be an equal number of standard deviations away from each mean, giving

$$T_i = \frac{\mu_i^{\text{in}} \sigma_i^{\text{out}} + \mu_i^{\text{out}} \sigma_i^{\text{in}}}{\sigma_i^{\text{out}} + \sigma_i^{\text{in}}}$$

This method of threshold determination assumes Gaussian distributions. Some previous studies by Buck [24], however, showed that the logarithms of average distortions are more nearly Gaussian than the distortions themselves; so the thresholds T_i were based on the statistics of the logarithms of distortions, instead of simply the distortion as shown in (2), (3), and (5).

(It was pointed out by a referee that using the Gaussian assumption is not necessary in determining the thresholds, and it may be part of the cause of a bias in the error rates toward false rejections that is reported in Sections III-D–F. The referee suggested using the empirical distributions obtained from the histograms of the true talker's distortions and of the imposter training data distortions to find the thresholds. This, we believe, would be a good way of determining the thresholds.)

To verify a speaker, the verification utterance $\mathbf{v}$ is coded in the appropriate codebook and the average log distortion is computed. This distortion value is compared to the threshold associated with that codebook. If the distortion value exceeds the threshold, the speaker is rejected; otherwise the speaker is accepted.

Preliminary experiments indicated that verification accuracy using a single verification utterance is poor [15]. To improve the verification accuracy, we based the verification decision on the results for several words. The next section describes our approach to extending this method to multiple words.

A. Multiple Word Approach

In previous work [15], we examined three ways of extending our method to more than one word. All three methods achieved about the same verification accuracy, and based on those results, we used the simplest of the three methods in this work.

A separate codebook is designed to represent each word that a speaker is required to say; if W words are spoken, there are W codebooks designed for each speaker. Next, for each speaker, separate thresholds are computed for each word, just as described in Section III. Now, an unknown speaker claims an identity and says the W words in the appropriate order. Each word is coded in the appropriate codebook, the distortion is compared to the threshold, and a verification decision is made for each word. For example, if a speaker is requested to say *zero* and *nine*, first the *zero* utterance is encoded with the *zero* codebook for that speaker and the resulting distortion is compared to the threshold; next the *nine* utterance is encoded with the *nine* codebook. To make a final verification decision, we use a majority rule; that is, the decision made in a majority of the individual word verification tests is used as the overall decision. In the case of ties, the speaker is rejected.

IV. Experiments

We first describe the speech databases and how the databases were partitioned for use in separate parameter studies and evaluation tests. We then list the parameters varied in the experiments. This is followed by three subsections; each subsection describes the verification results using one of the source models described in Section II.

A. Databases

We combined two databases to do these experiments, both collected by Texas Instruments Inc. (TI). The main difference in the databases is the resolution of the A/D converters. One database was digitized with a 12-bit converter; the second was digitized at a later time with a 16-bit converter. The differences in the two databases may bias the results slightly, but we do not believe the bias is significant.

Data for designing the codebooks to represent the speakers, determining the parameters for the in-class distributions, and testing verification accuracy, came from the database described in [25]. It contains 26 utterances of each digit (*zero* through *nine*) by 16 speakers (8 male and 8 female). We call this database TI-1. The database used for determining the parameters for the out-of-class distributions and for testing the imposter rejection capabilities of the methods contains two utterances of each of the 10 digits from 108 adult male and 112 adult female speakers [26]. (This database also contains 5 additional speakers that are in TI-1; these speaker's data were not used in our work.) This database is divided into two parts: a *training* part containing 54 male and 55 female speakers, and a *testing* part containing 54 male and 57 female speakers. We call this second database TI-2.

Automatic endpoint detection for both training and test utterances was used in our experiments. Our endpoint-detection algorithm is based on ideas presented in [27]

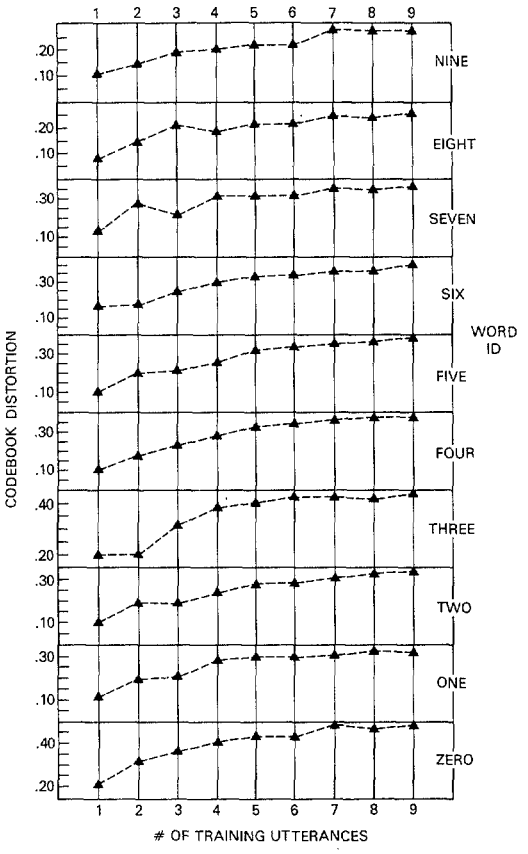


Fig. 1. The codebook design distortion for each of the 10 digits as a function of the number of codebook training utterances—one speaker's results.

and [28]. Briefly, the algorithm first analyzes the background noise to determine its average magnitude and then uses the result to set various thresholds that are used to find significant "energy clumps" in the data. See [9] for more details.

B. Database Partition

We first determined the number of training utterances required to model a speaker saying a digit. To do this, the average codebook-design distortion as a function of the number of training utterances was examined. For each speaker–digit combination, we designed a sequence of single-section codebooks, each containing 8 codewords. The first codebook was designed using a one-utterance training sequence, and each new codebook in the sequence was designed by increasing the number of training utterances by one. The average codebook-design distortion was recorded for each codebook. (See Fig. 1 for the results from a typical speaker.) We expected the average distortion to increase with each additional training utterance until the training data contained all the normal variations in a speaker's pronunciation of a word, and we then expected the distortion to stop increasing significantly with additional training data. We felt that a flattening of the distortion curve indicated that all the speaker's typical pronunciations were present. Fig. 1 shows that generally the distortion curve is flat by the 7th utterance. In addition, considering all 10 digits and the 16 speakers in TI-1, the average number of training utterances needed to reach 90 percent of the maximum codebook-design distortion was 8.

Based on the results of these experiments, in all three parameter studies described below, we designed digit codebooks for each speaker in the TI-1 database using the first 8 utterances of each digit. These 8 training utterances plus the next 4 utterances were used to estimate the parameters for the in-class distribution for each speaker–digit model. The next 7 digits in TI-1 were the speaker supplied verification data. For the parameter studies, the TI-2 *training* data were divided into two parts. One part containing the first 29 male and 30 female speakers was used to estimate the out-of-class distribution parameters; the second part containing the rest of the training portion of TI-2 (50 speakers) was used as imposter data.

Based on the results of the parameter studies, we chose several sets of words and codebook parameters to use in the full database tests of the three source models. In these tests, the training data for each speaker codebook again consisted of the first 8 utterances of a digit. The in-class parameter estimation data, however, consisted of the first 16 utterances. The remaining 10 utterances of each digit in TI-1 were the verification data. We used all 109 speakers in the training portion of TI-2 to estimate the out-of-class distribution parameters and the 111 speakers in the test portion as the imposters.

C. Experimental Parameters

We varied the codebook size—the number of codewords in a codebook—in the experiments. For single section codebooks, the codebook size is always a power of 2—i.e., $N = 2^R$, and we call R the codebook rate. For multisection codebooks, the size of each constituent section codebook is also a power of 2, and we call the section codebook rate R_S. Similarly, the size of matrix quantization codebooks is a power of 2, and we call the matrix codebook rate R_M.

In addition to the codebook rates, we varied the section length n for multisection codebooks and the matrix size K for matrix quantization codebooks during the parameters studies. The parameters used during verification always matched those used in designing the codebooks.

There are several parameters affecting the design of codebooks and thus the verification results that we did not vary. For one, after endpoint detection, we preprocessed the training and verification data by dividing each utterance into 24 equal length frames. This was done to provide a rough form of time normalization to all the utterances; previously, this was shown to help in isolated word recognition [16]. Also, since the spectra corresponding to nearly silent frames can be quite arbitrary and unspeechlike, to avoid cluttering codebooks with codewords that would result from including such frames, we used an energy threshold to ignore them. For single-section and multisection codebooks, we used an energy (sum-of-squares of data points) threshold of 250; this threshold was used both in codebook generation and speaker verification. For matrix quantization, we handled low-energy

TABLE I
SPEAKER VERIFICATION STUDY: SINGLE SECTION CODEBOOKS, MAJORITY RULE, AND ALL 10 DIGITS

Codebook Rate (R)	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections
1	800	7	112	15
2	800	1	112	7
3	800	0	112	7
4	800	0	112	6

TABLE II
SPEAKER VERIFICATION STUDY: A SINGLE CODEBOOK PER SPEAKER AND $R = 3$

Digit Spoken	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
ZERO	800	24 (3.0%)	112	10 (8.9%)	5.2
ONE	800	9 (1.1%)	112	19 (17.0%)	4.3
TWO	800	14 (1.8%)	112	10 (8.9%)	4.0
THREE	800	17 (2.1%)	112	17 (15.2%)	5.7
FOUR	800	12 (1.5%)	112	13 (11.6%)	4.2
FIVE	800	19 (2.4%)	112	26 (23.2%)	7.5
SIX	800	14 (1.8%)	112	17 (15.2%)	5.2
SEVEN	800	9 (1.1%)	112	16 (14.3%)	4.0
EIGHT	800	32 (4.0%)	112	15 (13.4%)	7.3
NINE	800	20 (2.5%)	112	20 (17.9%)	6.7

TABLE III
SPEAKER VERIFICATION RESULTS: $R = 3$, MAJORITY RULE, AND ALL 10 DIGITS

Speaker ID	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
TBS	222	0	10	0	0.0
WMF	222	0	10	0	0.0
RLD	222	1	10	0	0.0
GRD	222	1	10	1	2.1
KAB	222	5	10	2	6.7
MSW	222	0	10	0	0.0
REH	222	0	10	0	0.0
RGL	222	0	10	0	0.0
CJP	222	0	10	0	0.0
DFG	222	0	10	0	0.0
ALK	222	0	10	0	0.0
HNJ	222	0	10	0	0.0
GNL	222	0	10	1	0.0
JWS	222	0	10	0	0.0
SJN	222	0	10	1	0.0
SAS	222	0	10	1	0.0
Totals	3552	7 (0.2%)	160	6 (3.8%)	0.9

frames in the following manner. The first K-1 low-energy frames in a sequence were replaced with flat-spectrum frames with energy equal to 250; if more than K-1 frames occurred in a sequence, we ignored all but the first K-1. The reason for keeping the K-1 silent frames was to preserve the information in the transitions from silence-to-speech and vice versa, while eliminating any long all-silent training and verification segments.

D. Single Section Results

Parameter Studies: We varied the codebook rate R in these experiments. Verification decisions were made using all 10 digits and the majority-rule classifier, which was described in Section III-A. The results are listed in Table I for R ranging from 1 to 4. Most of the verification errors were false rejections. This implies that the thresholds T are too small. For R equal to 3 and 4, all errors were caused by just 2 of the 16 speakers.

For R equal to 3, we measured the verification accuracy of each digit individually; the results are in Table II. No single digit reliably verifies the speakers, and the individual digit results are also biased toward false rejections. The last column in Table II contains the square root of the product of the false-acceptance rate and the false-rejection rate ($\sqrt{FA * FR}$); this is considered a good overall performance measure [29].

Verification Tests: Because of the bias toward false rejections in the parameter study, we added more utterances to the data used in estimating the in-class distribution parameters; the new training set contained the 8 utterances used to design the codebooks and 8 additional utterances. (This was also done in the verification tests reported in Sections IV-E and F.) We felt that by including more utterances that were not in the codebook design set, the in-class distribution for a speaker would better represent new utterances from that speaker. Rate-3 codebooks were used in the verification tests because they did best in the parameter study and also because rate-3 codebooks yielded good speaker-trained isolated word recognition results [9].

The results, using all 10 digits in the verification decision, are listed by individual speaker in Table III. The majority of errors were caused by KAB and GRD; they also were the two difficult speakers in the parameter study. The results are still biased toward false rejections, how-

TABLE IV
SPEAKER VERIFICATION RESULTS: $R = 3$, MAJORITY RULE, AND 5 DIGITS

Digit Subset	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
01247	3552	19 (0.5%)	160	6 (3.8%)	1.4
35689	3552	27 (0.8%)	160	4 (2.5%)	1.4
25678	3552	25 (0.7%)	160	8 (5.0%)	1.9

TABLE V
SPEAKER VERIFICATION STUDY: MULTISECTION CODEBOOKS, MAJORITY RULE, ALL 10 DIGITS, 800 IMPOSTER ATTEMPTS, AND 112 ADMISSIBLE ATTEMPTS

Codebook Rate (R_S)	$n=12$		$n=8$		$n=4$		$n=2$		$n=1$	
	# FA	# FR	# FA	# FR	# FA	# FR	# FA	# FR	# FA	# FR
0	18	19	6	9	0	10	0	10	0	10
1	1	6	1	6	0	8	0	5	0	5
2	0	7	0	6	0	5	–	–	–	–

TABLE VI
SPEAKER VERIFICATION STUDY: A MULTISECTION CODEBOOK PER SPEAKER, $R_S = 2$, AND $n = 4$

Digit Spoken	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
ZERO	800	19 (2.4%)	112	12 (10.7%)	5.1
ONE	800	11 (1.4%)	112	13 (11.6%)	4.0
TWO	800	8 (1.0%)	112	16 (14.3%)	3.8
THREE	800	21 (2.6%)	112	18 (16.1%)	6.5
FOUR	800	13 (1.6%)	112	13 (11.6%)	4.3
FIVE	800	16 (2.0%)	112	31 (27.7%)	7.4
SIX	800	9 (1.1%)	112	15 (13.4%)	3.8
SEVEN	800	6 (0.8%)	112	24 (21.4%)	4.1
EIGHT	800	33 (4.1%)	112	9 (8.0%)	5.7
NINE	800	16 (2.0%)	112	19 (17.0%)	5.8

ever, the bias is smaller. $\sqrt{FA * FR}$ for this test was 0.9 percent.

Next several subsets of the digits were tested, each consisting of 5 digits. Based on the single digit parameter study, we used the best (0, 1, 2, 4, and 7), the worst (3, 5, 6, 8, and 9), and an arbitrary (2, 5, 6, 7, and 8) set of five digits. Results are listed in Table IV. Again, KAB and GRD were difficult speakers, but many other speakers contributed to the errors. Based on the $\sqrt{FA * FR}$ values for these tests, the verification accuracies obtained by representing each speaker by five words were significantly worse than those obtained using all 10 words. Generally, the degradation in performance was caused by additional false acceptances, and the overall performance was closer to the design goal of equal error rates for the two types of errors.

E. Multisection Results

Parameter Studies: We varied both the section length n and the section codebook rate R_S in these experiments. Table V shows the results. Generally for a fixed R_S value, better results are achieved using smaller n values. For $R_S = 2$, tests using n equal to 1 and 2 were not done because of insufficient codebook training data. Using $n = 4$ and $R_S = 2$, we tested the verification performance of the individual digits; the results are in Table VI. Again, as in the single section approach, no single digit gives good overall results and the errors are biased toward false rejections.

Verification Tests: We used $n = 4$ and $R_S = 2$ for the verification tests. These conditions were chosen because they both did well in the parameter study and in previous isolated word recognition work [16]. Table VII contains the results using all 10 digits to make the verification decision. No speaker was particularly difficult, as KAB and GRD were when using the single section approach, and, in general, the results are closer to the design goal of equal false-rejection and false-acceptance error rates than were the single section results. $\sqrt{FA * FR}$ was 0.6 percent.

Again, we did verification tests using the best (1, 2, 4, 6, and 7), the worst (0, 3, 5, 8, and 9) and an arbitrary (0, 1, 2, 3, and 4) set of five digits; the results are in Table VIII. The verification performance of the various five-digit subsets corresponded well with the expected performance based on the single digit study—i.e., the best five-digit set had the smallest $\sqrt{FA * FR}$, the worst set had the largest $\sqrt{FA * FR}$, and the arbitrary set had a $\sqrt{FA * FR}$ between the other two. The only consistently difficult speaker in these tests was KAB; averaged over the 3 five-digit tests, he had a false acceptance rate of 3.3 percent.

F. Matrix Quantization

Parameter Study: We varied the codebook rate R_m and the matrix size K in these experiments. For each K value, the maximum R_M was limited by the amount of codebook training data (poor codebooks often result if insufficient training data are used). The results are listed in Table IX. No obvious relationship between K and R_M is shown in these results. Using $R_M = 3$ and $K = 8$ (these conditions are also good for isolated word recognition [17]), we measured the verification performance of the individual

TABLE VII
SPEAKER VERIFICATION RESULTS: $R_S = 2$, $n = 4$, MAJORITY RULE, AND ALL 10 DIGITS

Speaker ID	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
TBS	222	1	10	0	0.0
WMF	222	0	10	0	0.0
RLD	222	0	10	0	0.0
GRD	222	1	10	0	0.0
KAB	222	2	10	0	0.0
MSW	222	0	10	0	0.0
REH	222	0	10	0	0.0
RGL	222	0	10	0	0.0
CJP	222	2	10	0	0.0
DFG	222	0	10	0	0.0
ALK	222	3	10	0	0.0
HNJ	222	0	10	0	0.0
GNL	222	0	10	2	0.0
JWS	222	0	10	0	0.0
SJN	222	0	10	0	0.0
SAS	222	0	10	0	0.0
Totals	3552	9 (0.3%)	160	2 (1.3%)	0.6

TABLE VIII
SPEAKER VERIFICATION RESULTS: $R_S = 2$, $n = 4$, MAJORITY RULE, AND 5 DIGITS

Digit Subset	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
12467	3552	26 (0.7%)	160	1 (0.6%)	0.7
03589	3552	34 (1.0%)	160	5 (3.1%)	1.7
01234	3552	17 (0.5%)	160	3 (1.9%)	0.9

TABLE IX
SPEAKER VERIFICATION STUDY: MATRIX QUANTIZATION CODEBOOKS, MAJORITY RULE, ALL 10 DIGITS, 800 IMPOSTER ATTEMPTS, AND 112 ADMISSIBLE ATTEMPTS

Codebook Rate (R_M)	K=4		K=8		K=12		K=24	
	# FA	# FR	# FA	# FR	# FA	# FR	# FA	# FR
2	6	11	5	16	4	14	2	8
3	0	10	0	8	0	9	–	–
4	0	8	–	–	–	–	–	–

TABLE X
SPEAKER VERIFICATION STUDY: A MATRIX QUANTIZATION CODEBOOK PER SPEAKER, $R_M = 3$, AND $K = 8$

Digit Spoken	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
ZERO	800	24 (3.0%)	112	12 (10.7%)	5.7
ONE	800	6 (0.8%)	112	14 (12.5%)	3.2
TWO	800	12 (1.5%)	112	16 (14.3%)	4.6
THREE	800	29 (3.6%)	112	21 (18.8%)	8.2
FOUR	800	12 (1.5%)	112	13 (11.6%)	4.2
FIVE	800	45 (5.6%)	112	33 (29.5%)	12.9
SIX	800	11 (1.4%)	112	16 (14.3%)	4.5
SEVEN	800	11 (1.4%)	112	22 (19.6%)	5.2
EIGHT	800	36 (4.5%)	112	14 (12.5%)	7.5
NINE	800	17 (2.1%)	112	22 (19.6%)	6.4

digits; these results are in Table X. As in the single section and multisection approaches, the errors are biased toward false rejections.

Verification Tests: The full database results using $R_M = 3$, $K = 8$, and all 10 digits are listed in Table XI. Once again, KAB was a difficult speaker. We tested the best (1, 2, 4, 6, and 7), the worst (0, 3, 5, 8, and 9), and an arbitrary (0, 1, 2, 3, and 4) five digits; the verification results are in Table XII. The relative performance of the five-digit sets did not correspond exactly with the expected results based on the individual digit performances, but the worst digit set did produce the poorest results.

V. SUMMARY AND DISCUSSION

The verification performances ($\sqrt{FA * FR}$) of the three source models when using only a single digit per speaker were similar—roughly varying from 4 to 8 percent depending on the digit, and consistently, the digits 1, 2, 4, and 7 individually did best in the speaker verification tests. When the individual digits were joined with the majority rule classifier, however, the verification performances of the three approaches were no longer equivalent. The multisection VQ source model did best when using the 10- and 5-digit sets of verification words (for the 10 digits, $\sqrt{FA * FR} = 0.6$ percent; for the best 5-digit set, $\sqrt{FA * FR} = 0.7$ percent). In addition, the multisection VQ approach came closer to satisfying the design goal of equal error rates, and the results on the 5-digit subsets corresponded more closely to the expected results (based on $\sqrt{FA * FR}$ for the individual digits). The next best source model was the single section approach, although

TABLE XI
SPEAKER VERIFICATION RESULTS: $R_M = 3$, $K = 8$, MAJORITY RULE, AND ALL 10 DIGITS

Speaker ID	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
TBS	222	1	10	0	0.0
WMF	222	0	10	0	0.0
RLD	222	0	10	1	0.0
GRD	222	1	10	1	2.1
KAB	222	3	10	3	6.4
MSW	222	0	10	0	0.0
REH	222	0	10	0	0.0
RGL	222	2	10	0	0.0
CJP	222	0	10	0	0.0
DFG	222	0	10	0	0.0
ALK	222	0	10	0	0.0
HNJ	222	0	10	0	0.0
GNL	222	0	10	2	0.0
JWS	222	0	10	0	0.0
SJN	222	0	10	1	0.0
SAS	222	0	10	1	0.0
Totals	3552	7 (0.2%)	160	9 (5.6%)	1.1

TABLE XII
SPEAKER VERIFICATION RESULTS: $R_M = 3$, $K = 8$, MAJORITY RULE, AND 5 DIGITS

Digit Subset	Number Of Imposter Attempts	False Acceptances	Number Of Admissible Attempts	False Rejections	$\sqrt{FA*FR}$ %
12467	3552	17 (0.5%)	160	8 (5.0%)	1.5
03589	3552	44 (1.2%)	160	9 (5.6%)	2.6
01234	3552	15 (0.4%)	160	6 (3.8%)	1.3

the differences in $\sqrt{FA * FR}$ values between the single section VQ (for the 10 digits, $\sqrt{FA * FR} = 0.9$ percent) and the MQ approach (for the 10 digits, $\sqrt{FA * FR} = 1.1$ percent) were small.

All three source models did well in the speaker verification tests. In retrospect, because all three use the same training data and contain essentially the same short-time spectrum information, simply stored differently, the similarity of their performance should not be surprising. At the heart of the three approaches is the clustering algorithm [12]; it designs all three codebook types. Both the multisection VQ and the MQ approach are straightforward generalizations of single section VQ that contain information about the time ordering of the speech spectra. One can see this by considering the multisection VQ approach with a section length n equal to the normalization length (24 in this study) and considering the MQ approach with the matrix size K equal to 1. Each approach reduces to single section VQ under the appropriate condition.

The single section VQ source model captures only the short-time spectrum shape information. This spectrum shape information is useful in speaker verification because it contains estimates of formant frequencies, relative formant amplitudes, and formant bandwidths, and these are correlated with the locations and physical sizes of the speech articulators. As such, the single section results are a measure of how well the short-time spectrum alone can characterize a speaker. In addition, because the codebook spectra are unordered, the single section VQ source model is directly applicable to text-independent speaker recognition, as pointed out by Soong [19] and Shikano [20].

It is generally believed, however, that examining parameters as a function of time is valuable in speaker verification for two reasons: 1) many of the speaker-characteristic properties of speech are the result of idiosyncrasies in the speaking habits of people, and 2) by considering the time sequence of parameters, the emphasis is on how the parameters vary rather than the exact value of a parameter. The multisection VQ and the MQ approaches are two different ways of incorporating some phonetic duration and coarticulation information into the verification process while maintaining the information-theoretic source model approach. Multisection VQ models a source by dividing it into several independent, time-ordered subsources. When used for speaker verification, a multisection VQ codebook provides an accurate representation of the speech spectra in an utterance, and multisection VQ coding enforces limits on the position of phonemes, the duration of phonemes, and to a lesser degree, the coarticulated phonemes in an utterance. Based on our results, adding this information improves the verification performance. As noted earlier, MQ models an utterance with a single codebook that contains an unordered set of time-ordered speech spectrum sequences. (Thus, MQ codebooks could be used in text-independent speaker recognition, just as single section codebooks are used.) These spectrum sequences correspond to stable continuant sounds or transitions from one sound to another. Thus, for large K values, MQ coding includes coarticulation and phonetic duration information. Our results show that MQ's use of this extra information does not improve the verification performance relative to the single section VQ results. This, of course, is contrary to our intuition, and it is unclear why the additional information inherent in the MQ approach does not improve the verification performance.

In addition to short-time spectra, a speaker will say an utterance with characteristic pitch [30] and stresses [1]. Because these are roughly independent of the spectrum

shape information, improvements in the verification accuracy could be achieved by adding pitch and short-time energy information to the verification process.

As an aside, it is interesting to consider how VQ speech coding could defeat a speaker verification or identification system. Our single section VQ results show that the source model found by using the Linde, Buzo, and Gray clustering algorithm [12] is an accurate representation of the short-time spectra produced by a speaker. Thus, to impersonate a speaker, one needs only to obtain training data spoken by that speaker and to design a VQ codebook with these data. Anyone could talk through this codebook (via VQ speech coding), and the resulting speech would be characteristic of the speaker who provided the training data. It seems this procedure would defeat any speaker recognition system that relies solely on short-time spectrum representations.

Finally, the connection between these speaker verification approaches and our previous isolated word recognition approaches needs emphasis. The parameters (codebook size, section length, and matrix size) and the source model (codebook) design procedure used in each of the speaker verification tests are exactly those used in our previous work on isolated word recognition [9], [16], [17]. In those studies, accuracies for speaker-trained recognition of the digits exceeded 99 percent, and the computational and memory storage requirements were shown to be quite small. The very good speaker verification and isolated word recognition results achieved using these approaches point toward a combined speaker–speech recognition system, and Shikano describes one example of how this could be done [20]. He proposes using VQ speaker recognition as a preprocessor for speaker-independent isolated word recognition. In his system, the speaker first is recognized using VQ codebooks, and then previously stored speaker-trained references are used to recognize the words. Thus, speaker-dependent recognition accuracies are obtained in a speaker-independent speech recognition system. In addition, however, adding VQ speech recognition to a speaker verification system should improve its performance. For example, suppose each speaker is represented by a set of speaker-trained VQ codebooks, as we described above. After a speaker claims an identity and wants to be verified, the verification system tells the speaker what words to say. The system first recognizes the words using the set of codebooks that were designed for the speaker whose identity was claimed and then compares the individual word distortions to the prestored verification thresholds. If the words are incorrectly recognized (which is much more likely to happen for an imposter) or the speaker fails the distortion tests, the speaker is rejected. This procedure would certainly reduce the number of false acceptances, although it may increase the number of false rejections.

Acknowledgment

I thank J. Buck for writing most of the software, R. Johnson for pointing out the VQ speaker impersonation scheme, and J. Shore for comments on this paper. In addition, I thank T. Schalk and G. Leonard for their help in obtaining the databases.

References

[1] R. C. Lummis, "Speaker verification by computer using speech intensity for temporal registration," *IEEE Trans. Audio Electroacoust.*, vol. AU-21, pp. 80–89, Apr. 1973.

[2] S. Pruzansky, "Pattern-matching procedure for automatic talker recognition," *J. Acoust. Soc. Amer.*, vol. 35, pp. 354–358, Mar. 1963.

[3] B. S. Atal, "Effectiveness of linear predictive characteristics of the speech wave for automatic speaker identification and verification," *J. Acoustic. Soc. Amer.*, vol. 55, pp. 1034–1312, 1974.

[4] W. S. Mohn, Jr., "Two statistical feature evaluation techniques applied to speaker identification," *IEEE Trans. Comput.*, vol. C-20, pp. 979–987, Sept. 1971.

[5] P. D. Bricker *et al.*, "Statistical techniques for talker identification," *Bell Syst. Tech. J.*, vol. 50, pp. 1427–1454, Apr. 1971.

[6] H. L. Van Trees, *Detection, Estimation, and Modulation Theory—Part 1.* New York: 1968.

[7] R. M. Gray, "Vector quantization," *IEEE ASSP Mag.*, pp. 4–29, Apr. 1984.

[8] 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.

[9] 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.

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

[11] 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, pp. 1075–1105, Apr. 1983.

[12] 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.

[13] S. Kullback, *Information Theory and Statistics.* New York: Dover, 1968; New York: Wiley, 1959.

[14] J. E. Shore and R. M. Gray, "Minimum-cross-entropy pattern classification and cluster analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-4, pp. 11–17, Jan. 1982.

[15] J. T. Buck, D. K. Burton, and J. E. Shore, "Text-dependent speaker recognition using vector quantization," in *Proc. 1985 ICASSP, IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 11.5.1–11.5.4.

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

[17] D. K. Burton, "Applying matrix quantization to isolated word recognition," in *Proc. ICASSP 1985, IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 1.8.1–1.8.4.

[18] K. P. Li and E. H. Wrench, Jr., "An approach to text-independent speaker recognition with short utterances," in *Proc. ICASSP 1983, IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, Apr. 1983, pp. 555–558.

[19] F. Soong *et al.*, "A vector quantization approach to speaker recognition," in *Proc. 1985 ICASSP, IEEE Int. Conf. Acoust., Speech, Signal Processing*, Tampa, FL, Mar. 1985, pp. 11.7.1–11.7.4.

[20] K. Shikano, "Text-independent speaker recognition experiments using codebooks in vector quantization," Carnegie-Mellon Univ., Apr. 1985, private communication (Abstract in program: 109th Meet. Acoust. Soc. Amer.).

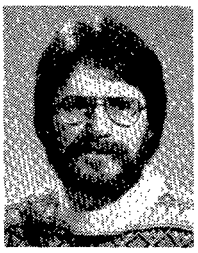
[21] C. Tsao and R. M. Gray, "Matrix quantizer design for LPC speech using the generalized Lloyd algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 537–545, June 1985.

[22] 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.

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

[24] J. T. Buck, "Vector quantization code book distortions as features

for maximum likelihood classification of isolated words," in *Proc. 1984 IEEE Global Telecommun. Conf. (GLOBECOM)*, Atlanta, GA, Nov. 1984, pp. 9.3.1-9.3.5.
[25] G. R. Doddington and T. B. Schalk, "Speech recognition: Turning theory to practice," *IEEE Spectrum*, vol. 18, pp. 26-32, Sept. 1981.
[26] R. G. Leonard, "A database for speaker-independent digit recognition," in *Proc. 1984 ICASSP, IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Diego, CA, Mar. 1984, pp. 42.11.1-42.11.4.
[27] 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.
[28] 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.
[29] G. R. Doddington, "Voice authentication gets the go-ahead for security systems," *Speech Technol.*, vol. 2, pp. 14-23, Sept./Oct. 1983.
[30] B. S. Atal, "Automatic speaker recognition based on pitch contours," *J. Acoust. Soc. Amer.*, vol. 52, pp. 1687-1697, 1972.



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, in 1974 and 1981, respectively.

Previously, he worked at Presearch, Inc., Crystal City, VA, on modeling the performance of antennas in cluttered environments; at Amecom, College Park, MD, on frequency agile transmitters; and at the Naval Research Laboratory, Washington, DC, on speech and signal processing. In 1985 he joined the Washington Research Laboratory of Entropic Processing Inc., Cupertino, CA, and is applying information-theoretic methods to signal processing, spectrum analysis, and speech coding.