AT&T Technical Journal
Vol. 64, No. 5, May–June 1985
Printed in U.S.A.

# Incorporation of Temporal Structure Into a Vector-Quantization-Based Preprocessor for Speaker-Independent, Isolated-Word Recognition

By A. F. BERGH, F. K. SOONG, and L. R. RABINER*

(Manuscript received November 15, 1984)

Recently a new structure for isolated-word recognition was proposed in which a separate Vector Quantization (VQ) code book was designed for each word in the vocabulary. The word-based VQs were used as a front-end preprocessor to eliminate word candidates whose distortion scores were large; a dynamic time-warping processor then resolved the choice among the remaining word candidates. The above scheme worked very well for small vocabularies; however, the major flaw was the lack of temporal information in the word-based VQ processor. As such, as the vocabulary grew in size and complexity, the ability of the VQ processor to resolve among similar sounding words decreased dramatically, and the effectiveness of the proposed recognition structure similarly decreased. To alleviate this difficulty a technique for incorporating temporal structure into the preprocessor is proposed. In particular, the probability density function of the time of occurrence for each vector in the code book is estimated from a training sequence. In the recognizer, the spectral distance score of the VQ is combined with a temporal distance score, for each frame in the word. An evaluation of the modified recognizer showed slightly improved performance on the digits vocabulary and greatly improved performance on a vocabulary of 129 airlines terms.

## I. INTRODUCTION

There has been a great deal of interest recently in isolated-word recognition techniques that maintain high performance, but do so at

* Authors are employees of AT&T Bell Laboratories.

low computational cost.[1-5] The reason for this renewed interest in "low-cost" recognizers is the desire to implement such systems on conventional microprocessors, where the computational power is nowhere near as great as needed for the "higher-cost" recognition systems.

One of the most promising of the low-cost recognizers is the Vector-Quantization (VQ)-based recognizer, originally proposed by Shore and Burton,[2] and modified by Burton et al.[4] and Pan et al.[5] The basic idea in this recognition system is to design a separate VQ code book for each word in the vocabulary, based on a training sequence of several tokens of each word by one or more talkers. In the original Shore and Burton implementation,[2] the recognizer chose the word in the vocabulary whose average quantization distortion (according to its particular code book) was minimum. In the implementation of Pan et al.,[5] the word-based VQs were used as a front-end preprocessor to eliminate word candidates whose distortion scores were large; a Dynamic Time Warping (DTW) processor then resolved the choice among the remaining word candidates.

Both of the above implementations of the word-based VQ recognizer worked very well for small vocabularies; however, as the vocabulary size and/or complexity grew, the ability of the VQ processor to resolve among similar sounding words decreased dramatically, and the effectiveness of the recognizer similarly decreased.

The major problem with the word-based VQ processor, for large vocabularies, was its inability to use temporal information, i.e., to integrate information about the times of occurrence of the speech sounds with the fact that the sounds occurred within the word. One simple method for incorporating this type of temporal information was proposed by Buzo et al.,[6] and developed by Burton et al.[4] In this approach, gross temporal information was incorporated into the recognizer by subdividing each input word into $R$ nonoverlapping regions, using a separate code book for each region. In this manner each word was characterized by $R$ code books, obtained from a training procedure in which a similar subdivision of each training word was made. Burton et al. reported good success with this method.[4]

An alternative procedure for incorporating temporal information into the VQ-based preprocessor is proposed in this paper. In particular, for each vector in each word-based code book, the probability density function of the time of occurrence (on a normalized time scale) is estimated from the same set of training sequences used to derive the code-book vectors. In the recognizer, the spectral distance score of the VQ preprocessor is combined with a (scaled) temporal distance score, for each frame in the word. We use the structure of a preprocessor to screen out unlikely word candidates (based on the combined spectral and temporal distance), and resolve the fine word distinctions with a DTW processor.

An evaluation of the modified recognizer structure, described above, was performed using both a small vocabulary (the 10 digits), and a moderate-size vocabulary (129 airline terms). Both vocabularies were tested in a speaker-independent mode, i.e., code books and probability histograms were generated from speaker-independent training sets. Results showed recognition error performance on both vocabularies was comparable to that of the best recognizers; however, computational costs were comparable to those of a "low-cost" recognizer.

The organization of this paper is as follows. In Section II we discuss the proposed recognition algorithm, which combines temporal information along with the spectral information of the word-based VQ preprocessor. In Section III we describe an experimental evaluation of the new recognition structure. In Section IV we review the results of the evaluation, and discuss potential ways of lowering the cost of the recognizer even further. Finally, in Section V, we summarize our findings.

## II. THE PROPOSED RECOGNITION ALGORITHM

A block diagram of the proposed recognizer is given in Fig. 1. The input speech signal is digitized at a 6.67-kHz rate, the word endpoints (beginning and ending frames) are detected, and a Linear Predictive Coding (LPC) analysis is performed on all frames within the word. The LPC analysis is an eighth-order analysis of 45-ms frames (300 samples), spaced every 15 ms (100 samples) along the word. Each overlapping 45 ms section of speech is windowed using a Hamming window, and an eighth-order autocorrelation analysis is performed (giving nine autocorrelation values per frame). The results of the LPC analysis are the set of frame log energies (suitably normalized to the peak log energy of the word), $E_i$, $1 \le i \le I$, and the LPC vectors $\mathbf{a}_i$, $1 \le i \le I$, where $I$ denotes the number of frames in the word.

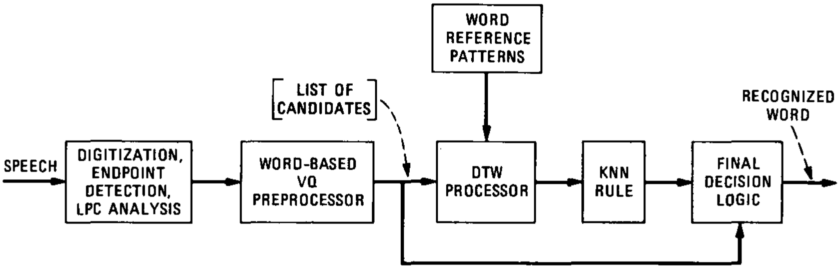

Fig. 1—Block diagram of isolated-word recognizer that incorporates a word-based vector quantization preprocessor and a dynamic time-warping processor.

The word-based LPC preprocessor uses the analysis results (i.e., the frame log energies and the LPC vectors) to eliminate all unlikely candidates from further analysis. Thus the output of the preprocessor is a list of candidates for the unknown word. A DTW processor then decides among the words in the candidate list using a conventional dynamic time-warping alignment of the unknown test word against a set of stored word reference patterns. A $K$ Nearest Neighbor (KNN) decision rule chooses the word whose average DTW distance of the $K$-best word patterns is smallest. In cases where the list of candidates from the preprocessor contains only a single choice, the DTW processor is bypassed and a final decision is made by the preprocessor.

### *2.1 The word-based VQ preprocessor*

A block diagram of the word-based VQ preprocessor is given in Fig. 2. Each word in the vocabulary is characterized, in the preprocessor, by a code book, $B$, and by a temporal probability table, $P$. The code book consists of a set of LPC vectors (supplemented by a log energy scalar), $\mathbf{b}_k$, $1 \le k \le L$, which characterize the LPC vectors of a training set of multiple occurrences of the word. The code-book vectors are chosen by a VQ design algorithm, which minimizes the average distortion between the training vectors and the code-book vectors.[7-9] Typically, for word recognition applications, values of $L$ (the total number of vectors in each word code book) range from 4 to 32.

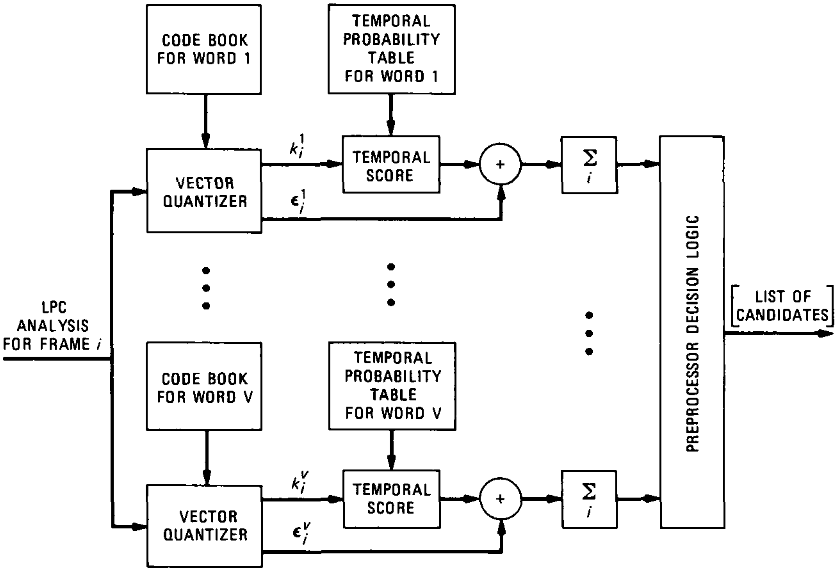

Fig. 2—Block diagram of the word-based vector quantization preprocessor, which combines spectral and temporal distance scores.

The temporal probability table, $P$, is derived from both the code book, $B$, and the word training data in the following way. The elements of $P$ are the values $p_k(t)$, defined as:

$p_k(t) =$ probability that the code-book vector, $k$, occurs at normalized time $t = i/I$ within the word.

Thus the values $p_k(t)$ (where suitably quantized values of $t$ are used in practice) constitute a temporal probability table for the code-book vectors. The way in which values of $p_k(t)$ are obtained, from the training set, is as follows:

1. Each training sequence is linearly warped to a fixed length, $\hat{I} =$ 40 frames. (Thus values of $p_k(t)$ are obtained for $t = 1/40, 2/40, \cdots,$ 40/40.)

2. Each vector of each linearly warped training sequence is vector quantized, using code book $B$.

3. At each time $t$, all code-book vectors whose spectral distortion distance score is within a fixed threshold, $\Delta$, of the minimum distortion score for the frame are considered to have occurred.

4. The value used for $p_k(t)$ is the ratio between the number of times code-book vector $k$ occurred at time $t$ (as defined in step 3 above), and the number of times any code-book vector occurred at time $t$, over the entire training set for the word. In this manner $\sum_{k=1}^{L} p_k(t) = 1$ for all $t$.

To illustrate the results of the above procedure, Fig. 3 shows the resulting $p_k(t)$ temporal probability tables for an $L = 8$ vector code book for the word six with a training set of 150 tokens of the word derived from 150 different talkers (75 male, 75 female). A value of $\Delta = 0.25$ was used in computing $p_k(t)$. Experimentation with $\Delta$ showed the resulting temporal probability tables were insensitive to $\Delta$ over a broad range; this was because with $L = 8$ (or 16) vectors, generally there was only a small fraction of the code-book vectors whose distortion scores were low. Given a large enough training set, the exact value of $\Delta$ (as long as it was relatively small) is almost irrelevant.

The temporal probability tables of Fig. 3, for the word six, show that a smooth probability density was obtained for all vectors. Further, we see that for some vectors a unimodal distribution resulted; for other vectors distinct multimodal distributions are found. In this example, the code-book vectors whose sounds represent the vowel /I/ have a unimodal distribution, since this sound occurs only at a single place in the word six. Code-book vectors whose sounds represent the fricative /s/ have a distinct two-mode distribution, since /s/ occurs at both the beginning and end of the word six. Finally, code-book vectors whose sounds represent silence have three modes, since silence can be

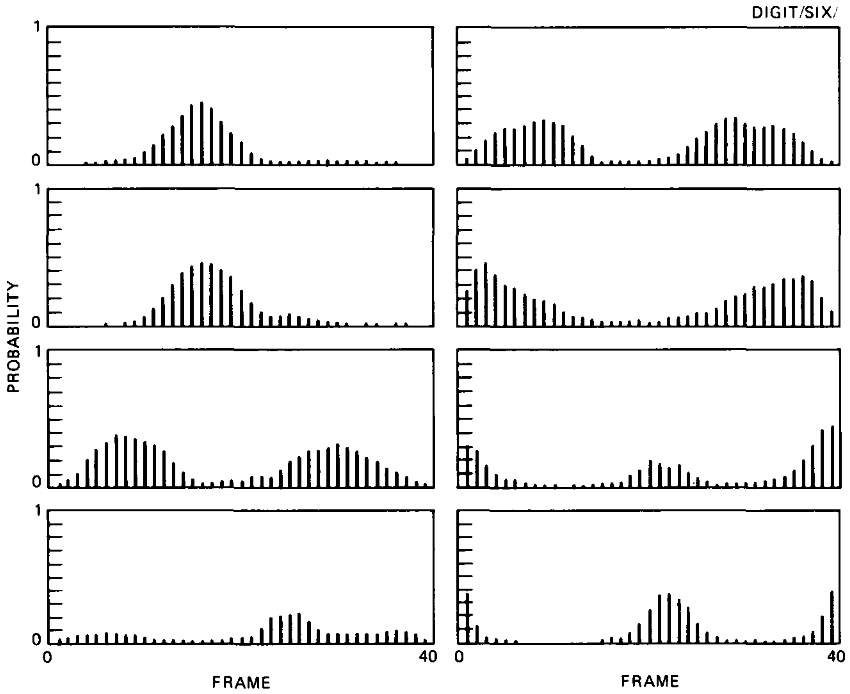

Fig. 3—Estimates of temporal probability density functions for the eight code-book vectors of the digit six. (*Forty-frame normalization of the word duration is assumed.*)

found at the beginning, end, and in the stop gap of the word six. All three types of distributions are clearly seen in the data of Fig. 3.

For convenience, and to reduce computation, the temporal probability tables were stored as

$$\hat{p}_k(t) = -\gamma \log[p_k(t)], \tag{1}$$

i.e., as negative log probabilities, so they could be combined readily with the LPC distances. The multiplier, $\gamma$, was chosen so that, averaged over the entire training set, the average value of $\hat{p}_k(t)$ was the same as the average LPC distance. Typically, the value of $\gamma$ was about 0.45 for $L = 8$ vector code books, and about 0.22 for $L = 16$ vector code books. Also, values of $p_k(t)$ were clipped at a level of $10^{-4}$; hence no temporal probability score was 0.

### *2.2 Combining LPC distance and temporal probability score*

After a great deal of investigation into ways of combining LPC distance and temporal probability scores, the resulting distance score that was used was

$$d(\mathbf{a}_i, E_i, B, P) = (1 - \alpha)d_{\text{SP}}(\mathbf{a}_i, E_i, B) + \alpha d_{\text{TP}}(k_i, P), \tag{2}$$

where $d_{\mathrm{SP}}$ was the spectral (LPC combined with energy) distance and $d_{\mathrm{TP}}$ was the temporal probability distance. The scaling value $\alpha$ was chosen by optimization and determined the mix of spectral and temporal "distances." A value of $\alpha = 0$ represents pure spectral distance; similarly, a value of $\alpha = 1.0$ represents pure "temporal distance."

The spectral distance, which combined the LPC distance with the energy distance, had the form

$$d_{\mathrm{SP}}(\mathbf{a}_i, E_i, B) = \min_{1 \le k \le L} [d_{\mathrm{LPC}}(\mathbf{a}_i, \mathbf{b}_k) + cf(d_E(E_i, \hat{E}_k))], \tag{3}$$

where

$$d_{\mathrm{LPC}}(\mathbf{a}_i, \mathbf{b}_k) = \left( \frac{\mathbf{b}_k' V_{\mathbf{a}_i} \mathbf{b}_k}{\mathbf{a}_i' V_{\mathbf{a}_i} \mathbf{a}_i} - 1 \right), \tag{4}$$

with $V_{\mathbf{a}_i}$ being the autocorrelation matrix of the input frame, $E_i$ being the normalized log energy of the input frame, and $\hat{E}_k$ being the normalized log energy of the $k$th code-book vector. We then have

$$d_E(E_i, \hat{E}_k) = |\hat{E}_k - E_i|, \tag{5}$$

with

$$f(E) = \begin{cases} 0 & 0 \le E \le E_{\mathrm{LO}} \\ E - E_{\mathrm{LO}} & E_{\mathrm{LO}} < E \le E_{\mathrm{HI}} \\ E_{\mathrm{HI}} - E_{\mathrm{LO}} & E_{\mathrm{HI}} < E, \end{cases} \tag{6}$$

where $c$, $E_{\mathrm{LO}}$, $E_{\mathrm{HI}}$, and $E_{\mathrm{OF}}$ were suitably chosen constants. (We used $c = 0.1$ $E_{\mathrm{LO}} = 6$ dB, and $E_{\mathrm{HI}} = 20$ dB.)

The temporal distance of eq. (2) was of the form

$$d_{\mathrm{TP}}(k_i, P) = \hat{p}_k([i \mid I]), \tag{7}$$

where $[i \mid I]$ is the rounded value of $i/I$ to the nearest 1/40.

The following sequence of steps was required to generate a combined distance score in the preprocessor:

1. Vector quantize the input frame (by each word-based code book), at time $t = i/I$, consisting of LPC vector $\mathbf{a}_i$ and normalized energy $E_i$, and determine the minimum spectral distance, $d_{\mathrm{SP}}$, and the index of the best code-book vector, $k_i$.

2. Access the temporal distance as $\hat{p}_{k_i}(t)$, where $t$ is quantized to the nearest 1/40 (since tables with 40 entries were used).

3. Combine $d_{\mathrm{SP}}$ and $d_{\mathrm{TP}}$ according to eq. (2).

The above procedure is performed at each frame for each word in the vocabulary, and the resulting distance scores are accumulated for each word, as shown in Fig. 2.

The preprocessor decision logic is essentially the same as used by Pan et al.,[5] namely:

1. Find all word candidates $v$, such that the average distortion, $D^v$,

$$D^v = \frac{1}{I} \sum_{i=1}^{I} d(\mathbf{a}_i, E_i, B^v, P^v) \tag{8}$$

is within a fixed threshold, $\delta$, of the minimum average distortion across all words.

2. If only a single word candidate exists, then the recognition is over—i.e., no DTW processing is required.

3. If more than one word candidate exists, then use the DTW processor to make the final recognition decision among the word candidates.

We now describe the results of a series of experiments designed to evaluate the performance of the overall recognizer of Figs. 1 and 2.

## III. EXPERIMENTAL EVALUATION

Two databases were used to evaluate the performance of the recognizer. All recordings were made over a standard, local, dialed-up telephone line. The first database was a digits set consisting of four sets of 1000 digits each (100 talkers·10 digits/talker). We call the digits sets DIG1, DIG2, DIG3, and DIG4. Their characteristics are as follows:

DIG1—100 talkers (50 male, 50 female), 1 replication of each digit by each talker.[10] These recordings have been used as a training set in a wide variety of evaluations of isolated-word recognizers.

DIG2—Same 100 talkers and recording conditions as DIG1; recordings made several weeks later than those of DIG1.

DIG3—100 new talkers (50 male, 50 female), 1 averaged occurrence of each digit by each talker obtained from averaging a pair of robust tokens of the digit.[11,12] The transmission conditions (i.e., analog front end, filter cutoff frequencies, etc.) differed slightly from those used in recording the DIG1 and DIG2 databases.

DIG4—A second group of 100 new talkers (50 male, 50 female), 20 recordings of each digit by each talker.[13] A random sampling of 1 of the recordings of each digit by each talker was used. The transmission conditions differed substantially from those used in recording the other databases.

The templates (12 per word, speaker independent) for the DTW processing were created from the data of set DIG1. The training data for the word-based VQ preprocessor (to get the code books, $B^v$, and the temporal probability tables, $P^v$) were derived from a randomly chosen set of 150 tokens of each word from sets DIG1, DIG3, and DIG4. (Of course these same training data could have been used to create the speaker-independent reference templates for DTW processing; however, a conveniently available template set was used.) For testing the recognizer, all four digit sets were used.

The second database was a vocabulary of 129 words used in an airlines information and reservation system.[14] Two sets of data, called AIR1 and AIR2, were used. Their characteristics were:

AIR1—100 talkers (50 male, 50 female), 1 averaged occurrence of each word by each talker obtained from averaging a pair of robust tokens of the word.[12]

AIR2—20 new talkers (10 male, 10 female), 1 replication of each word by each talker. The data of set AIR1 were used to create both the word reference templates (speaker independent, 12 per word), and to give the word code books and word temporal probability tables. The data of set AIR2 were used to test the recognizer.

### *3.1 Results on the digits vocabulary*

For each of the digit test sets, a preliminary test run was performed in which the preprocessor was used by itself to make the final recognition decision based on the word with the lowest combined spectral plus temporal distance score. (Equivalently, $\delta$, in the decision logic, was set to 0.) The distance combining parameter, $\alpha$, in eq. (2) was then varied from 0 to 1 (in steps of 0.1) and a curve of the preprocessor recognition accuracy versus $\alpha$ was computed. A typical such curve for the test set DIG1 is given in Fig. 4. The behavior of the recognition rate, shown in this figure, is typical for all the digit test sets. It can be seen that for $\alpha = 0$ (only spectral distance) and for $\alpha = 1.0$ (only temporal distance), the recognition rate of the preprocessor (91.4 percent for $\alpha = 0$, 91.2 percent for $\alpha = 1.0$) is significantly lower than its value at the peak of the curve (97.5 percent for $\alpha = 0.7$). This result strongly points out the value of combining spectral and temporal

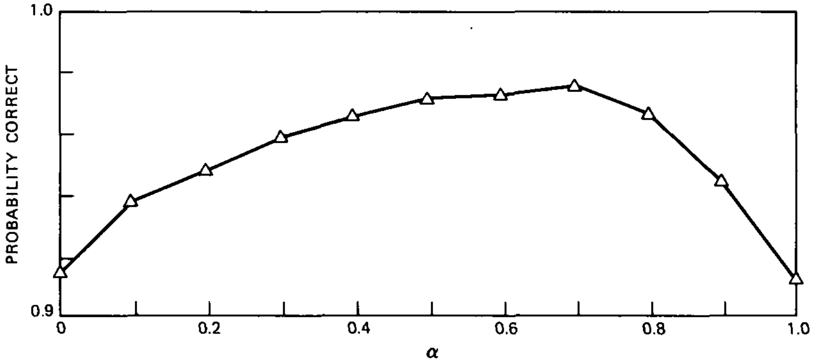

Fig. 4—Curve of average digit recognition rate versus the combining multiplier, $\alpha$, for the data of test set DIG1. (Note that $\alpha = 0$ corresponds to pure spectral distance and $\alpha = 1.0$ corresponds to pure temporal distance.)

Table I—Average error rates for digits vocabulary

| Code-Book Size | Average Digit Error Rate (%) | | | | |
|---|---|---|---|---|---|
| | DIG1 | DIG2 | DIG3 | DIG4 | Overall |
| | | | (a) Processor Alone | | |
| 8 | 2.5 | 3.1 | 3.1 | 4.3 | 3.3 |
| 16 | 2.5 | 2.6 | 2.5 | 3.7 | 2.8 |
| | | | (b) Complete Recognizer | | |
| 8 | 2.9 | 2.5 | 2.2 | 2.9 | 2.6 |
| 16 | 1.3 | 2.3 | 2.2 | 2.8 | 2.2 |

distances in the preprocessor. It also can be seen that in the vicinity of the peak (near $\alpha = 0.7$), the recognition rate is fairly constant (its value at $\alpha = 0.5$ is 97.1 percent); hence, a fairly broad region of choices for $\alpha$ is possible. Across the four digit test sets, the optimum value of $\alpha$ varied from 0.4 to 0.7. If we used the value $\alpha = 0.5$ for all digit sets, the preprocessor recognition rate changed less than 0.2 percent, on average.

A complete set of performance results on the digits test sets is given in Table I. Table Ia gives average digit error rates for the preprocessor working without the DTW processor, for the four test sets (and an overall average), for code books with 8 and 16 vectors per word. The average digit error rate is 3.3 percent for 8 vector code books, and 2.8 percent for 16 vector code books. Table Ib gives average digit error rates for the complete recognizer, as a function of code-book size. The threshold, $\delta$, in the preprocessor was set so that, on average, about 83 percent of the time no DTW was required (i.e., the preprocessor made the final decision), and about 17 percent of the time, the average number of word candidates passed on to the DTW processor was 2.25. No quantization of the reference templates in the DTW processor was used; previous experience with this data set indicates that no degradation need occur if the reference template quantization is done correctly.[5]

From Table Ib it can be seen that the entire recognizer achieved an average digit error rate of 2.6 percent for $L = 8$ vector code books, and 2.2 percent for $L = 16$ vector code books. These results represent improvements of 0.6 to 0.7 percent in word accuracy; for 4000 test digits, such a result is statistically significant.

### *3.2 Results on the airline vocabulary*

For the airline vocabulary, a curve of preprocessor average performance versus the combining multiplier $\alpha$ was again run, and the results are given in Fig. 5. Although the form of the curve is similar to that of the digits case (Fig. 4), performance improves significantly when

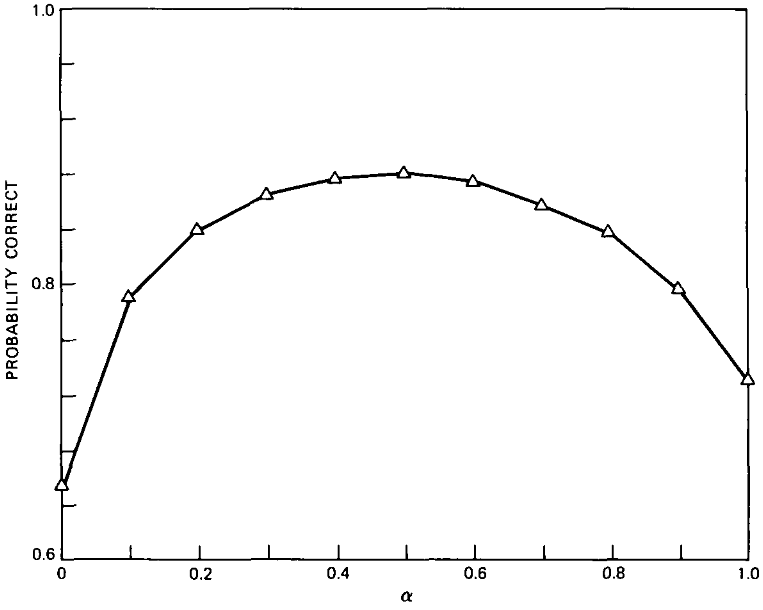

Fig. 5—Curve of average word recognition rate versus the combining multiplier, $\alpha$, for the airline test data.

Table II—Average word error rates for the airlines vocabulary

| Code-Book Size | Average Word Error Rate (%) | |
|---|---|---|
| | Preprocessor Alone | Total Recognizer |
| 8 | 14.8 | 11.7 |
| 16 | 11.9 | 8.9 |

using both spectral and temporal distance, as opposed to either spectral or temporal distance alone. We see from Fig. 5 that for $\alpha = 0$ (spectral distance only), the preprocessor achieves a 65.4-percent accuracy; for $\alpha = 1.0$ (temporal distance only), the accuracy is 73.2 percent (it is better than the result for $\alpha = 0$). However, for $\alpha = 0.5$, the combined distance yields a performance of 88.1-percent word accuracy, an improvement in accuracy of from 15.5 to 22.7 percent over the individual distances.

The overall recognizer performance on the airline vocabulary is given in Table II. The 8-vector-per-word system has a preprocessor error rate of 14.8 percent, whereas the 16-vector-per-word system has a preprocessor error rate of 11.9 percent. By setting the preprocessor decision threshold so that a unique decision was made by the preprocessor on 76 percent of the trials, and on 24 percent of the trials, an average of 2.5 candidates (out of 129 possible) were passed on to the DTW processor, the overall word error rates fell to 11.7 percent for the 8-vector code books, and to 8.9 percent for the 16-vector code books.

### *3.3 Typical recognition example*

To illustrate how the addition of temporal information aids the preprocessor, Fig. 6 shows a recognition case in which a word (the

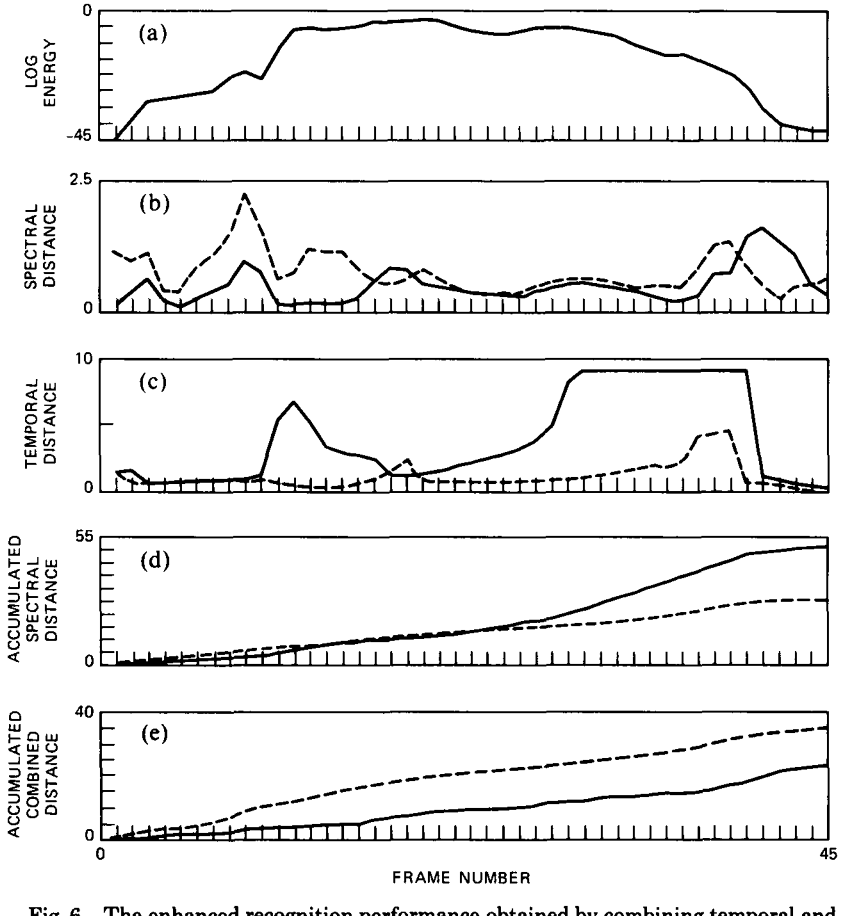

Fig. 6—The enhanced recognition performance obtained by combining temporal and spectral distances in the preprocessor: (a) the test word (zero) log energy contour; (b) spectral and (c) temporal distances on a frame-by-frame basis (solid curve is for the word three, dashed curve is for the correct word zero); (d) accumulated spectral and (e) accumulated combined distance scores.

digit zero) would have been misrecognized (as the digit three) based on VQ spectral distances alone, but is correctly recognized based on the combination of spectral and temporal distance. Shown in this figure are the log energy contour of the test word zero (Fig. 6a), the VQ spectral distance (frame by frame) for both the word zero (dashed line) and the word three (solid line, Fig. 6b), the log probability distances for both words (Fig. 6c), the accumulated spectral distance scores for both words (Fig. 6d), and the combined, accumulated total distance scores for both words (Fig. 6e). On the basis of VQ spectral matches, the preprocessor would have made a hard error since the distance for zero was not close enough to the distance for three; however, using the combined distance the correct word zero was uniquely recognized. The reason that the temporal distance helped so much, in this case, was the large temporal distance during the /O/ vowel in zero for the word three. Thus, although there is a code-book vector that matches the /O/ spectrum well in three, the probability of it occurring at the end of the word is very small. Although this example is an extreme case, it does illustrate well why the addition of temporal information to the preprocessor can help the performance to improve.

## IV. DISCUSSION

The results presented in the previous section clearly show that the addition of temporal information to a word-based VQ preprocessor increases the accuracy of the recognizer and makes it more robust to vocabulary size and complexity.

To gain perspective on how the current system performance compares with previous recognizers, Table III gives digit recognition error rates for the current system, for the best DTW recognizer,[15] and for a previous recognizer using a word-based VQ preprocessor.[5] Similarly, Table IV gives word recognition error rates, for the airline vocabulary, for the current system, for the best DTW recognizer, and for a previous recognizer using a word-based VQ preprocessor.[5]

For the digits, the DTW system performs slightly better, on average, than the current system. However, the best performance is on test sets DIG1 and DIG2, from which the word reference templates were derived. On the test sets DIG3 and DIG4, the current system performed

Table III—Average digit error rates for three recognition systems

| | Average Digit Error Rate (%) | | | | |
|---|---|---|---|---|---|
| Recognizer | DIG1 | DIG2 | DIG3 | DIG4 | Overall |
| Current system | 1.3 | 2.3 | 2.2 | 2.8 | 2.2 |
| DTW alone | 0.0 | 0.6 | 2.7 | 3.9 | 1.8 |
| Previous recognizer | — | 2.0 | — | — | — |

Table IV—Average word error rates for three recognition systems for the airline vocabulary

| Recognizer | Average Word Error Rate (%) |
|---|---|
| Current system | 8.9 |
| DTW alone | 10.2 |
| Previous recognizer | 12.6 |

slightly better than the DTW recognizer. On the test set DIG2 (which was the only common one between the current system and the previous recognizer with the preprocessor), the system performances were essentially the same.

For the airline vocabulary we see that the error rate of the current system is 1.3 percent lower than that of the DTW recognizer alone, and 3.7 percent lower than the previous recognizer based on the VQ preprocessor. For this vocabulary a real performance improvement has been achieved.

### *4.1 Computational considerations*

It remains for us to show that this increase in system performance is achieved at essentially no increase in system cost (i.e., computational complexity). To do this we define the following system variables:

$L$ = Code-book size
$V$ = Vocabulary size
$I$ = Average number of frames in a word
$Q$ = Number of templates per word in DTW
$p$ = LPC order
$\gamma$ = Average fraction of words that are resolved in the preprocessor
$\beta$ = Average fraction of words passed on to DTW processor, when more than a single word candidate exists.

The computation of the preprocessor can be expressed as:

$$C_{\text{PRE}} = V \cdot I \cdot L \cdot (p + 1) \quad *, +$$

and the computation of the DTW postprocessor is

$$C_{\text{POST}} = (1 - \gamma)\beta C_{\text{DTW}},$$

where

$$C_{\text{DTW}} = V \cdot Q \cdot \frac{I^2}{3}(p + 1) \quad *, +.$$

The overall computation of the recognizer is

$$\begin{aligned} C_R &= C_{\text{PRE}} + C_{\text{POST}} \\ &= V \cdot I \cdot (p+1) \left( L + Q\,(1-\gamma)\,\beta\,\frac{I}{3} \right). \end{aligned}$$

The ratio between the full DTW computation (without a preprocessor) and the current recognizer computation is then

$$R = \frac{C_{\text{DTW}}}{C_R} = \frac{Q(I/3)}{L + Q(1-\gamma)\beta\left(\frac{I}{3}\right)}.$$

Substituting typical values of $Q = 12$, $I = 40$, $p = 8$, $L = 8$ (or 16), $(1 - \gamma) = 0.25$, $\beta = 0.02$, we get

$$\begin{aligned} R &\cong 20 \qquad (L = 8) \\ &\cong 10 \qquad (L = 16). \end{aligned}$$

Thus, a computational reduction (over a standard DTW recognizer) of from 10 to 20 times is achieved by the proposed recognizer.

### *4.2 Further computational reduction via universal code book*

Although the performance of the proposed recognizer is impressive, it is possible to reduce its computational complexity even further. If we analyze the computation above, the major computation is in the preprocessor, where a total of $V \cdot L$ dot product distances need to be computed for each test frame. In the case where $V$ is large (e.g., the 129-word airline vocabulary), the total number of code-book vectors becomes large. In such a case it would be less expensive to use a universal code book (word and talker independent) of say 1024 vectors, and to choose the word-based code books from the universal code book. In this manner the number of distance computations per frame is fixed, and does not grow with the vocabulary size $V$. Of course, it must be shown that performance will not degrade, but it seems reasonable that for a sufficiently large code book, this will indeed be the case.

## V. SUMMARY

In this paper we have shown how the addition of temporal information into the preprocessor of an isolated-word recognizer can improve the system performance and make the overall recognizer more robust to vocabulary size and complexity. The way in which the temporal information was added was straightforward; namely, we defined and measured from a training set a probability density function on the time of occurrence of the code-book vectors in the word-based VQ preprocessor. A temporal distance was defined as the scaled, negative log probability of the probability of occurrence of the vector chosen by the vector quantizer. A combined measure in which the spectral distance (from the VQ) was added to the temporal distance was used in the recognizer and shown to improve performance for both a digits and moderate-size airline vocabulary. Finally, it was shown that, on average, the computational complexity of the resulting recognizer was less than that required for a conventional dynamic time-warping implementation by at least a factor of ten, whereas the recognition performances of the two systems were comparable.

## REFERENCES

1. K. Shikano, "Spoken Word Recognition Based Upon Vector Quantization of Input Speech," Trans. Comm. Speech Res. (December 1982), pp. 473–80.
2. J. E. Shore and D. K. Burton, "Discrete Utterance Speech Recognition Without Time Alignment," IEEE Trans. Inform. Theory, *IT-29*, No. 4 (July 1983), pp. 473–91.
3. 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," B.S.T.J., *62*, No. 4 (April 1983), pp. 1075–105.
4. D. K. Burton, J. T. Buck, and J. E. Shore, "Parameter Selection for Isolated Word Recognition Using Vector Quantization," Proc. ICASSP 84, San Diego, Calif. (March 1984), pp. 9.4.1–4.
5. K. C. Pan, F. K. Soong, and L. R. Rabiner, "A Vector Quantization Based Preprocessor for Speaker Independent Isolated Word Recognition," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-33*, No. 3 (June 1985).
6. A. Buzo, H. G. Martinez, and C. Riviera, "Discrete Utterance Recognition Based Upon Source Coding Techniques," Proc. ICASSP 82, Paris, France (May 1982), pp. 539–42.
7. Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization," IEEE Trans. Commun., *COM-28*, No. 1 (January 1980), pp. 84–95.
8. B. Juang, D. Wong, and A. H. Gray, Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-30*, No. 2 (April 1982), pp. 294–303.
9. L. R. Rabiner, M. M. Sondhi, and S. E. Levinson, "A Vector Quantizer Combining Energy and LPC Parameters and Its Application to Isolated Word Recognition," AT&T Bell Lab. Tech. J., *63*, No. 5 (May–June 1984), pp. 721–35.
10. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-27*, No. 4 (August 1979), pp. 336–49.
11. L. R. Rabiner and J. G. Wilpon, "A Simplified Robust Training Procedure for Speaker Trained, Isolated Word Recognition Systems," J. Acoust. Soc. Amer., *68*, No. 5 (November 1980), pp. 1271–5.
12. J. G. Wilpon, L. R. Rabiner, and A. Bergh, "Speaker Independent Isolated Word Recognition Using a 129-Word Airline Vocabulary," J. Acoust. Soc. Amer., *72*, No. 2 (August 1982), pp. 390–6.
13. A. E. Rosenberg, K. L. Shipley, and D. E. Bock, "A Speech Data Base Facility Using a Computer Controlled Cassette Tape Deck," J. Acoust. Soc. Amer., Suppl. 1, *72*, (Fall 1982), p. 580.
14. S. E. Levinson, and K. L. Shipley, "A Conversational Mode Airline Information and Reservation System Using Speech Input and Output," B.S.T.J., *59*, No. 1 (January 1980), pp. 119–37.
15. J. G. Wilpon and L. R. Rabiner, "A Modified K-Means Clustering Algorithm for Use in Speaker Independent Isolated Word Recognition," IEEE Trans. Acoust., Speech, and Signal Processing, *ASSP-33*, No. 3 (June 1985).

## AUTHORS

**Arthur F. Bergh,** currently enrolled at Swarthmore College; AT&T Bell Laboratories, 1980—. Mr. Bergh has been engaged in speech-recognition research in the Acoustics Research Department.

**Lawrence R. Rabiner,** S.B. and S.M., 1964, Ph.D., 1967 (Electrical Engineering), The Massachusetts Institute of Technology; AT&T Bell Laboratories, 1962—. From 1962 through 1964 Mr. Rabiner participated in the cooperative plan in electrical engineering at AT&T Bell Laboratories, in Whippany and Murray Hill, New Jersey. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently, Mr. Rabiner is engaged in research on speech communications and digital signal processing techniques. He is coauthor of *Theory and Application of Digital Signal Processing* (Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice-Hall, 1983). Member, National Academy of Engineering, Eta Kappa Nu, Sigma Xi, Tau Beta Pi. Fellow, Acoustical Society of America, IEEE.

**Frank K. Soong,** B.S., 1973, National Taiwan University, M.S., 1977, University of Rhode Island, Ph.D., 1983, Stanford University, all in Electrical Engineering; AT&T Bell Laboratories, 1982—. From 1973 to 1975 Mr. Soong served as a teacher at the Chinese Naval Engineering School at Tsoying, Taiwan. In 1982 he joined the technical staff at AT&T Bell Laboratories, where he engaged in research in speech coding and recognition. Member, IEEE.