An Efficient Vector-Quantization Preprocessor for Speaker Independent Isolated Word Recognition

K. C. Pan
F. K. Soong
L. R. Rabiner
A. F. Bergh

AT&T Bell Laboratories
Murray Hill, New Jersey 07974

ABSTRACT. Recently a new structure for isolated word recognition was proposed based on the ideas of vector quantization (VQ). In this scheme a separate VQ codebook, for each word in the vocabulary, was designed, based on a training sequence of several tokens of each word by one or more talkers. In the original implementation, the recognizer chose the word in the vocabulary whose average quantization distortion (according to its particular codebook) was minimum. In the proposed implementation, the word-based VQ's are used as a front end preprocessor to eliminate word candidates whose distortion scores are large; a DTW processor then resolves the choice among the remaining word candidates (i.e. those which are passed on by the preprocessor). Both of the above schemes work very well for small vocabularies; however the major flaw is the lack of temporal information in the word-based VQ processor. As such, as the vocabulary for recognition grows in size and complexity, the ability of the VQ processor to resolve among similar sounding words decreases dramatically, and the effectiveness of the proposed recognition structure similarly decreases. To alleviate this difficulty a technique for incorporating temporal structure into the preprocessor is also proposed. In particular, the probability density function of the time of occurrence for each vector in the codebook is estimated from the same training sequence used to derive the codebook vectors. In the recognizer, the spectral distance score of the VQ is combined with a (scaled) temporal distance score, for each frame in the word. An evaluation of the proposed recognizer showed good performance on both the digits vocabulary, and on a vocabulary of 129 airlines terms.

I. Introduction

There has been a great deal of interest, recently, in techniques for isolated word recognition which maintain high performance, but do so at 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]. The basic idea in this recognition system is to design a separate VQ codebook 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 codebook) was minimum. This 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 [5].

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, non-overlapping, regions, and using a separate codebook for each region. In this manner each word was characterized by R codebooks, 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 implementing a word-based VQ recognizer which incorporates temporal information is proposed in this paper. In particular, for each vector in each word-based codebook, 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 codebook 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. The word-based VQ preprocessor screens out unlikely word candidates (based on the combined spectral and temporal distance), and is followed by a DTW processor which resolves fine word distinctions.

An evaluation of the proposed recognizer structure 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. codebooks 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.

II. The Proposed Recognition Algorithm

A block diagram of the proposed recognizer is given in Figure 1. The input speech signal is digitized at a 6.67 kHz rate, the word endpoints (beginning and ending frames) are detected, and an LPC analysis is performed on all frames within the word. The LPC analysis is an 8th order analysis of 45 msec frames (300 samples), spaced every 15 msec (100 samples) along the word. Each overlapping 45 msec section of speech is windowed using a Hamming window, and an 8th order autocorrelation analysis is performed (giving 9 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, $1 \leqslant i \leqslant I$, and the LPC vectors $\mathbf{a}_i$, $1 \leqslant i \leqslant I$, where I denotes the number of frames in the word.

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 by a conventional dynamic time warping alignment of the unknown test word against a set of stored word reference patterns. A 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.

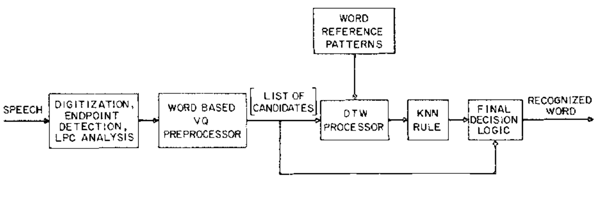

Fig. 1 Block diagram of isolated word recognizer.

CH2118-8/85/0000-0874 $1.00 © 1985 IEEE

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 codebook, B, and by a temporal probability table, P. The codebook consists of a set of LPC vectors (supplemented by a log energy scalar), $\mathbf{b}_k$, $1 \leqslant k \leqslant L$, which characterize the LPC vectors of a training set of multiple occurrences of the word. The codebook vectors are chosen by a VQ design algorithm which minimizes the average distortion between the training vectors and the codebook vectors [7-9]. Typically, for word recognition applications, values of L (the total number of vectors in each word codebook) range from 4 to 32.

The temporal probability table, P, is derived from both the codebook, 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 codebook 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 codebook 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 $t = 1/40, 2/40, \ldots, 40/40$).
2. Each vector of each linearly warped training sequence is vector quantized, using codebook B.
3. At each time t, all codebook vectors whose distortion distance score is within a fixed threshold, Δ, 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 codebook vector k occurred at time t (as defined in step 3 above), and the number of times any codebook 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.

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 readily be combined with the LPC distances. The multiplier γ, 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 γ was about 0.45 for $L = 8$ vector codebooks, and about 0.22 for $L = 16$ vector codebooks. Also values of $p_k(t)$ were clipped at a level of 10^{-4}; hence no temporal probability score was 0.

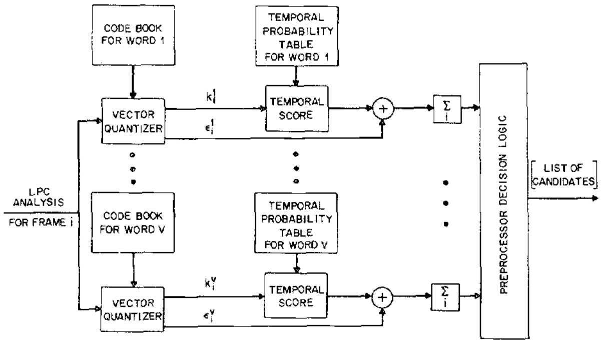

Fig. 2 Block diagram of the word-based vector quantization preprocessor.

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

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

where d_{SP} was the spectral (LPC combined with energy) distance and d_{TP} was the temporal probability distance. The scaling value α 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

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

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} codebook vector. We then have

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

with

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

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

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

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

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

The preprocessor decision logic is as follows:

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, δ, 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.

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 4 sets of 1000 digits each (100 talkers × 10 digits/talker). We call the digits sets DIG1, DIG2, DIG3 and DIG4. 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 codebooks, 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. For testing the recognizer, all 4 digit sets were used.

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

AIR 1 — 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.

AIR 2 — 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 codebooks 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, δ, in the decision logic, was set to 0). The distance combining parameter, α, in Eq. (2) was then varied from 0 to 1 (in steps of 0.1) and a curve of the preprocessor recognition accuracy versus α was computed. A typical such curve for the test set DIG1 is given in Figure 3. 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% for $\alpha = 0$, 91.2% for $\alpha = 1.0$) is significantly lower than its value at the peak of the curve (97.5% for $\alpha = 0.7$). This result strongly points out the value of combining spectral and temporal distances in the preprocessor. It can also 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%); hence a fairly broad region of choices for α is possible. Across the 4 digit test sets, the optimum value of α varied from 0.4 to 0.7. If we used the value $\alpha = 0.5$ for all digit sets, the preprocessor recognition rate changes less than 0.2%, 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 4 test sets (and an overall average), for codebooks with 8 and 16 vectors per word. Table Ib gives average digit error rates for the complete recognizer, as a function of codebook size. The threshold, δ, in the preprocessor was set so that, on average, about 83% of the time no DTW was required (i.e. the preprocessor made the final decision), and about 17% 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].

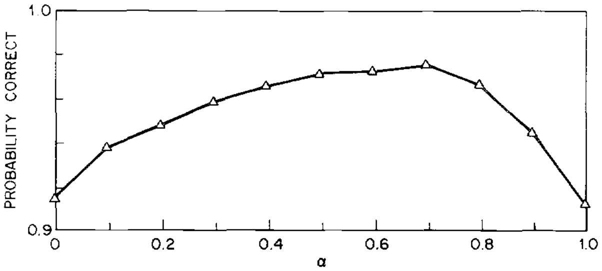

Fig. 3 Curve of average digit recognition rate versus the combining multiplier, α.

3.2 Results on the Airline Vocabulary

For the airline vocabulary, a curve of preprocessor average performance versus the combining multiplier α was again run, and the results are given in Figure 4. Although the form of the curve is similar to that of the digits case (Fig. 3), the level of improvement in performance by using both spectral and temporal distance; over either spectral or temporal distance alone, is indeed impressive. We see From Fig. 4 that for $\alpha = 0$ (spectral distance only), the preprocessor achieves a 65.4% accuracy; for $\alpha = 1.0$ (temporal distance only), the accuracy is 73.2% (it is better than the result for $\alpha = 0$). However, for $\alpha = 0.5$, the combined distance yields a performance of 88.1% word accuracy, an improvement in accuracy of from 15.5% to 22.7% 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%, whereas the 16 vector per word system has a preprocessor error rate of 11.9%. By setting the preprocessor decision threshold so that a unique decision was made by the preprocessor on 76% of the trials, and on 24% 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% for the 8 vector codebooks, and 8.9% for the 16 vector codebooks.

Codebook Size	Average Digit Error Rate (%)				
	DIG1	DIG2	DIG3	DIG4	Overall
8	2.5	3.1	3.1	4.3	3.3
16	2.5	2.6	2.5	3.7	2.8

Codebook Size	Average Digit Error Rate (%)				
	DIG1	DIG2	DIG3	DIG4	Overall
8	2.9	2.5	2.2	2.9	2.6
16	1.3	2.3	2.2	2.8	2.2

Table I. Average Digit Error Rate of;
(a) Preprocessor alone and; (b) Complete Recognizer

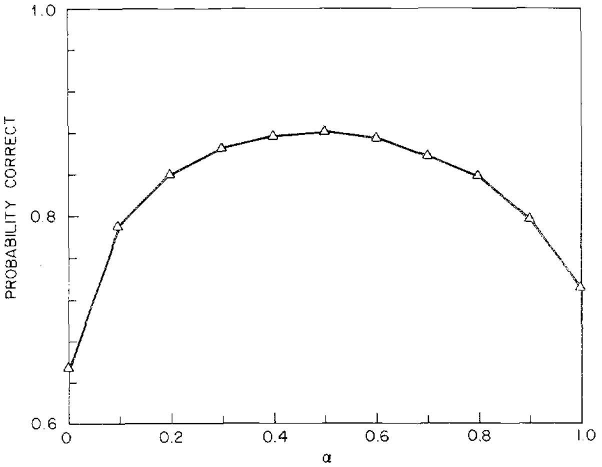

Fig. 4 Curve of average word recognition rate versus the combining multiplier, α, for the airline test data.

Codebook Size	Average Word Error Rate (%)	
	Preprocessor Alone	Total Recognizer
8	14.8	11.7
16	11.9	8.9

Table II. Results for the Airlines Vocabulary

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 IIIa gives digit recognition error rates for the current system and for the best DTW recognizer. Similarly, Table IIIb gives word recognition error rates, for the airline vocabulary, for the current system and for the best DTW recognizer.

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 slightly better than the DTW recognizer.

For the airline vocabulary we see that the error rate of the current system is 1.3% lower than that of the DTW recognizer alone.

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 = codebook size, V = vocabulary size, I = average number of frames in a word, Q = number of templates per word in DTW, p = LPC order, γ = average fraction of words which are resolved in the preprocessor, β = 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:

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

and the computation of the DTW postprocessor is

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

where

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

The overall computation of the recognizer is

$$C_R = C_{\mathrm{PRE}} + C_{\mathrm{POST}} = V \cdot I \cdot (p+1)(L+Q(1-\gamma)\,\beta\,\frac{I}{3})$$

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

$$R = \frac{C_{\mathrm{DTW}}}{C_R} = \frac{Q(I/3)}{L+Q(1-\gamma)\,\beta\,(\frac{I}{3})}$$

Substituting typical values of $Q = 12$, $I = 40$, $p = 8$, $L = 8$ (or 16), $(1-\gamma) = 0.25$, $\beta = 0.02$, we get $R \cong 20$ for $(L=8)$, or $R \cong 10$ for $(L=16)$. 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 Codebook

Although the performance of the proposed recognizer is impressive, it is possible to reduce its computational complexity even further in the following way. As seen from the analysis of 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 codebook vectors becomes large. In such a case it would be less expensive to use a universal codebook (word and talker independent) of say 1024 vectors, and to choose the word-based codebooks from among the universal codebook. 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 codebook, this will indeed be the case.

Recognizer	Average Digit Error Rate (%)				
	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

Recognizer	Average Word Error Rate (%)
Current System	8.9
DTW Alone	10.2

Table III. Average Word Error Rates for (a) Digits and; the Airline Vocabulary

REFERENCES

[1] K. Shikano, "Spoken Word Recognition Based Upon Vector Quantization of Input Speech," *Trans. of Committee on Speech Research*, pp. 473-480, Dec. 1982.

[2] J. E. Shore and D. K. Burton, "Discrete Utterance Speech Recognition Without Time Alignment," *IEEE Trans. on Information Theory*, Vol. IT-29, No. 4, pp. 473-491, July 1983.

[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," *Bell Syst. Tech. J.*, Vol. 62, No. 4, pp. 1075-1105, April 1983.

[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, CA, pp. 9.4.1-9.4.4, March 1984.

[5] K. C. Pan, F. K. Soong, and L. R. Rabiner, "A Vector Quantization Based Preprocessor for Speaker Independent Isolated Word Recognition," submitted for publication.

[6] A. Buzo, H. G. Martinez, and C. Riviera, "Discrete Utterance Recognition Based Upon Source Coding Techniques," *Proc. ICASSP 82*, Paris, France, pp. 539-542, May 1982.

[7] Y. Linde, A. Buzo, and R. M. Gray, "An Algorithm for Vector Quantization," *IEEE Trans. on Communications*, Vol. COM-28, No. 1, pp. 84-95, Jan. 1980.

[8] B. Juang, D. Wong, and A. H. Gray Jr., "Distortion Performance of Vector Quantization for LPC Voice Coding," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 2, pp. 294-303, April 1982.

[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 Laboratories Tech. J.*, Vol. 63, No. 5, pp. 721-735, May-June 1984.