379
N81d
No. 2910

SPEECH RECOGNITION USING A SYNTHESIZED CODEBOOK

DISSERTATION

Presented to the Graduate Council of the
University of North Texas in Partial
Fulfillment of the Requirements

For the Degree of

DOCTOR OF PHILOSOPHY

By

Lloyd A. Smith, B.M., M.S., M.S.

Denton, Texas

August, 1988

Smith, Lloyd A., Speech Recognition Using a Synthesized Codebook.
Doctor of Philosophy (Computer Science), August, 1988, 164 pp., 37
tables, 33 illustrations, bibliography, 86 titles.

Speech sounds generated by a simple waveform synthesizer were used to create a vector quantization codebook for use in speech recognition. Recognition was tested over the TI-20 isolated word data base using a conventional DTW matching algorithm. Input speech was band limited to 300 - 3300 Hz, then passed through the Scott Instruments Corp. Coretechs process, implemented on a VET3 speech terminal, to create the speech representation for matching. Synthesized sounds were processed in software by a VET3 signal processing emulation program. Emulation and recognition were performed on a DEC VAX 11/750.

The experiments were organized in 2 series. A preliminary experiment, using no vector quantization, provided a baseline for comparison.

The original codebook contained 109 vectors, all derived from 2 formant synthesized sounds. This codebook was decimated through

the course of the first series of experiments, based on the number of times each vector was used in quantizing the training data for the previous experiment, in order to determine the smallest subset of vectors suitable for coding the speech data base. The second series of experiments altered several test conditions in order to evaluate the applicability of the minimal synthesized codebook to conventional codebook training.

The baseline recognition rate was 97%. The recognition rate for synthesized codebooks was approximately 92% for sizes ranging from 109 to 16 vectors. Accuracy for smaller codebooks was slightly less than 90%. Error analysis showed that the primary loss in dropping below 16 vectors was in coding of voiced sounds with high frequency second formants. The 16 vector synthesized codebook was chosen as the seed for the second series of experiments.

After one training iteration, and using a normalized distortion score, trained codebooks performed with an accuracy of 95.1%. When codebooks were trained and tested on different sets of speakers, accuracy was 94.9%, indicating that very little speaker dependence was introduced by the training.

ACKNOWLEDGEMENTS

I would like to express my appreciation to all of the people who have supported my work; in particular, I would like to thank my colleagues at Scott Instruments Corp., where my doctoral research was carried out.

Mark Newell designed the SI2010 signal processing chip and wrote the code which emulated it on the Scott Instruments research computer system. Lisan Lin wrote the graphics routines used to display speech representations and implemented the recognition test environment. Gary Goodman helped with mathematical aspects of speech synthesis and, over the years, has tried to teach me to discipline my thinking. Brian Scott is responsible for the ideas underlying my research, and, as friend and mentor, he has shown me that science is, above all, a creative endeavor.

Finally, I would like to thank my wife, Carolyn, for help with the manuscript, and, especially, for her patience through years of night classes and weekend work sessions.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iii
LIST OF TABLES	v
LIST OF ILLUSTRATIONS	viii
Chapter	
1. INTRODUCTION	1
2. VECTOR QUANTIZATION IN SPEECH RECOGNITION	19
3. SOFTWARE TOOLS	43
4. DESIGN AND METHODOLOGY	55
5. RESULTS OF EXPERIMENTS USING SYNTHESIZED CODEBOOKS	79
6. RESULTS OF EXPERIMENTS USING TRAINED CODEBOOKS...	102
7. CONCLUSIONS	113
APPENDIX	
A. RECOGNITION ACCURACY BY SPEAKER	124
B. CONFUSION MATRICES	140
BIBLIOGRAPHY	156

LIST OF TABLES

Table	Page
1. Recognition accuracy using synthesized codebooks	80
2. Confusion matrix for baseline experiment	93
3. Confusion matrix for 109 vector codebook	94
4. Confusion matrix for 16 vector codebook	99
5. Confusion matrix for 14 vector codebook	100
6. Recognition accuracy using trained codebooks	104
7. Confusion matrix for one pass training	110
8. Speaker scores for baseline experiment	125
9. Speaker scores for 109 vector codebook	126
10. Speaker scores for 64 vector codebook	127
11. Speaker scores for 48 vector codebook	128
12. Speaker scores for 32 vector codebook	129
13. Speaker scores for 24 vector codebook	130
14. Speaker scores for 18 vector codebook	131
15. Speaker scores for 16 vector codebook	132
16. Speaker scores for 14 vector codebook	133

Table	Page
17. Speaker scores for 12 vector codebook	134
18. Speaker scores for one pass training	135
19. Speaker scores for recursive training	136
20. Speaker scores for one sided vq	137
21. Speaker scores for normalized distortion	138
22. Speaker scores for cross speaker training	139
23. Confusion matrix for baseline experiment	141
24. Confusion matrix for 109 vector codebook	142
25. Confusion matrix for 64 vector codebook	143
26. Confusion matrix for 48 vector codebook	144
27. Confusion matrix for 32 vector codebook	145
28. Confusion matrix for 24 vector codebook	146
29. Confusion matrix for 18 vector codebook	147
30. Confusion matrix for 16 vector codebook	148
31. Confusion matrix for 14 vector codebook	149
32. Confusion matrix for 12 vector codebook	150
33. Confusion matrix for one pass training	151
34. Confusion matrix for recursive training	152

Table	Page
35. Confusion matrix for one sided vq	153
36. Confusion matrix for normalized distortion	154
37. Confusion matrix for cross speaker training	155

LIST OF ILLUSTRATIONS

Figure	Page
1. Discriminative network	2
2. Phonetic coding of speech	5
3. Vector quantization	7
4. Conventional speech recognizer with vector quantization . .	20
5. Bakis model	21
6. Word based vq recognizer	24
7. Synthesized /IY/	47
(a) waveform	
(b) magnitude spectrum	
8. Recognizer continuity constraints	49
9. Recognition window	52
10. Speech preprocessing	58
11. Speech processing of vowel "oh"	60
(a) sampled waveform	
(b) differentiated	
(c) infinitely peak clipped	
12. Endpoint detection using correlation envelope	63
13. F1 - F2 coverage in 109 vector synthesized codebook	66

Figure	Page
14. Vector representation of word "one"	69
(a) original	
(b) quantized by 109 vector codebook	
15. System diagram -- synthesized codebook experiments	70
16. System diagram -- trained codebook experiments	76
17. Vectors sorted by use -- 109 vector codebook	82
18. Vectors sorted by use, voiced vs. unvoiced -- 109 vector codebook	83
19. Vectors sorted by use -- 32 vector codebook	84
20. Vectors sorted by use -- 16 vector codebook	85
21. Vectors sorted by use -- 14 vector codebook	86
22. F1 - F2 coverage in 32 vector codebook	87
23. F1 - F2 coverage in 16 vector codebook	88
24. F1 - F2 coverage in 14 vector codebook	89
25. Mean distortion by codebook size	90
26. Variability of distortion scores by codebook size	91
27. Vector representation of word "three"	101
(a) original	
(b) quantized by 16 vector codebook	
(c) quantized by 14 vector codebook	
28. Vectors sorted by use -- single pass trained codebooks . . .	105

Figure	Page
29. Vectors sorted by use -- recursively trained codebooks . . .	105
30. Vector match changes during training	108
31. Mean distortion during codebook training	108
32. Vector representation of word "one"	109
(a) original	
(b) quantized by one pass trained codebook	
33. Selected results from vector quantization studies	115

CHAPTER 1

INTRODUCTION

The work described here is an investigation into the use of synthesis techniques in speech coding for recognition. This chapter discusses some of the problems faced in encoding speech, defines a particular coding method known as vector quantization, and gives the reasons for choosing vector quantization as the coding method for the investigation.

Background

Early research in automatic speech recognition was concerned with deriving an optimal representation of the speech signal. During the ARPA (Advanced Projects Research Agency) Speech Understanding Project in the seventies, attention turned to more powerful methods of pattern matching with the development of dynamic time warping and markov models (Klatt, 1977). As these techniques have matured, more emphasis has again been placed on speech signal representations.

Ideally, a speech representation should be compact and it should possess the "discriminative network" capability (Moore et al., 1983) -- that is, parts of words that sound similar should have a single representation in the vocabulary or lexicon. This provides a means for focusing a recognizer's attention on the appropriate parts of similar sounding words. Consider a spelunking vocabulary containing the words "stalagmite" and "stalactite." A traditional dynamic time warping (DTW) recognizer, using a separate template for each vocabulary word, weights its recognition scores equally over the entire utterance -- but that is inappropriate in this case, when superficial differences in the "stala-" and "-ite" portions of the words may overpower the differences between the short "-gm-" and "-ct-" segments. The discriminative network represents identical parts of different words by common paths through the vocabulary network (Figure 1, after Moore et al.).

Figure 1. Discriminative network.

Current commercial speech recognizers function without either of these attributes, but, as vocabulary sizes grow, the need for both compact representations and focused matching becomes critical. Some researchers have attempted to use strings of phonetic labels to represent speech (Olson and Belar, 1956). This would be ideal, since all English words can be represented using about 40 phonemes (linguists differ on the exact number). By contrast, about 250 allophones, 1200 diphones, or 4500 demisyllables would be required (Silvertsen, 1961). Virtually all of English is encompassed by about 4400 syllables, but complete coverage may require as many as 20,000 (Lee and Lochovsky, 1983).

Unfortunately, simple phonetic labels have been unsuccessful as a representation for recognition because there is no unique acoustic manifestation of a phoneme. One reason for acoustic variability is noise -- differences in room acoustics, background environment, telephone line characteristics, and so forth; another is dialectal or spectral differences between voices, or even in the same voice at different times. These problems may be solved by modeling the noise (Roe, 1987; Compernelle, 1987) or by adapting to the speaker

(Bonneau and Gauvain, 1987; Golibersuch, 1983; Di Benedetto and Lanaro, 1984), but a more basic problem lies in the way that speech is acoustically encoded -- there is no simple correspondence between phonemes and segments of the acoustic signal. The acoustic cues for one phoneme may overlap those for the phonemes on either or both sides of it. Figure 2, after Liberman (1970), shows how the phonemes in the word "bag" are acoustically encoded (the heavy dark lines represent the first and second formants). The cues for /B/ and /G/ both overlap those for /AE/, and the cues for /AE/ extend over the entire word.

The complex nature of the phonetic coding of speech means that each phoneme has a number of different acoustic manifestations which are dependent on the phonetic environment; consequently, it is very difficult to automatically label the phonemes with a high degree of accuracy.

Figure 2. Phonetic coding of speech.

With the inherent difficulty in using phonetic labels, it becomes necessary to find an alternative speech representation which is still compact and holds the potential of forming discriminative networks for matching. Fant (1973) suggests modeling speech by "sound segments," a sound segment being a section of the speech signal characterized by its sound features (Chomsky and Halle, 1968). The sound segment is a pragmatic rather than a linguistic unit; it is based on the acoustic signal and claims no psychological relevance. Several

adjacent sound segments may carry information concerning one phoneme, and one sound segment may simultaneously carry information on several phonemes. A typical utterance is expected to possess more sound segments than phonemes, but the exact number depends on the acoustic signal and on the discriminability of the sound feature system. The compactness of a sound segment representation, then, is determined partly by the signal itself, and partly by the implementor's choice of sound feature system. Clearly, sound segments hold the potential for forming discriminative networks, since similar segments in different words (or within the same word) can be replaced by one prototype segment.

There are many ways that a sound segment representation of speech might be implemented, but one method that has become increasingly popular in recent years is known as vector quantization (Gersho and Cuperman, 1983).

Vector Quantization

Shannon laid the foundation for vector quantization by proving that any signal can be encoded at its theoretically minimal bit rate by collecting the samples into groups, or vectors, and encoding the

vectors rather than individual samples (Shannon, 1948). Lloyd (1957) took the first step toward practical implementation of Shannon's model when he designed a clustering algorithm for generating a "codebook" containing representative signal samples. Recently, Linde et al. (1980) extended Lloyd's algorithm to vector quantization. Figure 3 shows a block diagram of a vector quantizer.

Figure 3. Vector quantization.

Vectors from an unknown pattern are mapped to the codebook by finding the codebook vector that is most similar to the pattern vector, based on some distance or "distortion" measure. The codebook index of the chosen vector is then transmitted or otherwise used to represent the pattern vector. Decoding is simply a matter of retrieving the indexed vector from the decoder's copy of the

codebook.

The algorithm developed by Linde, Buzo, and Gray is known as the generalized Lloyd or LBG algorithm. Since its publication, vector quantization has been applied to speech representations ranging from prediction coefficients (Buzo et al., 1980) to cochleagrams (Loeb and Lyon, 1987), and to problems as diverse as low bit rate speech transmission (Reininger and Wolf, 1985), formant tracking (Kopec, 1986), and speaker identification (Burton, 1987).

Use of Vector Quantization by Speech Recognizers

Vector quantization is easily applied in recognition because recognizers process speech in 10 - 30 millisecond (ms) "frames," where each frame is represented by a vector. The vectors are derived by different methods in different systems; they may represent measurements of particular speech features (such as voicing and amplitude) but more often they are coefficients derived from a signal processing operation such as linear predictive coding (LPC) or FFT. In speech terms, the vector components represent the speech features, the codebook vectors represent the sound segments, and the distortion measure represents the similarity between an

input frame of speech and a given sound segment. The quantization may be "one sided" or "two sided"; in both cases the recognizer's vocabulary is vector quantized, but in one sided vector quantization, input (or "unknown") utterances are matched directly against the vocabulary. In two sided vector quantization, the unknown is vector quantized before matching.

In addition to its adaptability to frame based systems, vector quantization has attracted attention as a speech representation for the two reasons discussed above. First, it is compact; one system is able to maintain a 20,000 word vocabulary using just 100 vectors of 20 parameters each (IBM Speech Recognition Group, 1987). Second, it holds the potential for forming discriminative networks.

To see how a discriminative network is created in vector quantization, consider again Moore's example, illustrated in Figure 1. Ideally, similar sounding sections of different words will be represented by the same vectors; thus, the "stala-" and "-ite" sections of the two words should be represented by a common set of codebook vectors. The difference in recognition scores between these two words and an unknown utterance, then, would be

determined solely by the "-gm-" and "-ct-" segments. The discriminative network property is guaranteed, to a certain extent, by vector quantization, as similar vectors of input speech map to the same vector. The network is much more powerful, however, if it is linguistically relevant -- that is, if it is phonetically rather than acoustically based; an extra step of processing, using phonological rules, may be able to accomplish this in a vector quantization system.

Problems With Coding Speech Using Vector Quantization

In spite of its advantages, vector quantization does have drawbacks as a technique for speech coding. Since the codebook is dependent on the specific data used in its creation, it is generally speaker and vocabulary dependent. This has been turned to an advantage in speaker identification (Soong et al., 1987), but in speech recognition and transmission, it means that a new codebook must be generated and/or transmitted whenever the speaker or vocabulary changes. Some attempts to overcome these limitations include statistical manipulation of the signal (Reininger and Wolf, 1985), and development of universal codebooks through training on many speakers (Furui, 1987).

A second problem with vector quantization is that codebook generation algorithms are only locally optimal, and depend on an initial, or "seed" codebook (Parsons, 1987); in fact, the most difficult task in a conventional vector quantization algorithm is the generation of an initial codebook (Equitz, 1987).

Research Goal

The purpose of this thesis was to investigate the viability of using speech synthesis techniques to create a universal (speaker independent) vector quantization codebook.

While synthesis has not been used in vector quantization, it is not a radically new concept. Klatt (1974) proposed a model for word verification in which hypothesized words would be synthesized and matched against the acoustic input. Thosar and Rao (1976) and Paliwal and Rao (1982) have used a synthesizer to model transitional segments in a speech recognition system. A number of methods for estimating formant frequencies have depended on analysis by synthesis (Parsons, 1987), wherein the vocal tract transfer function is modeled by synthesis, and the resulting spectrum is matched against that of the acoustic input. A logical extension is to apply

synthesis techniques to vector quantization.

The development of a synthesized codebook is attractive for a number of reasons. First, such a codebook would not have to be trained for each new user or application -- it would be independent of speaker and vocabulary; alternatively, a synthesized codebook could be used as a universal seed codebook for training in speech applications. Second, while the codebook would not by itself allow speaker independent recognition, it would be an important step in that direction, since it could provide speaker independence at the lowest level of speech representation. Third, such a codebook would make possible speaker independent low-bit rate speech transmission, with synthesis directly from the codebook vectors or from the waveform segments that were used to generate the vectors (Roucos and Wilgus, 1985). Finally, a synthesized codebook could conceivably be used for speech analysis -- phonetic labeling, for example, or formant analysis in a method similar to that of Heinz and Stevens (1961) or Bell et al. (1961); the results could be used by a recognizer based on a phonetic grammar or on speech features.

Organization of the Thesis

Chapter two reviews the literature in vector quantization, emphasizing its use in speech recognition. Chapter three describes the major software tools used in the study -- the synthesizer used to create sounds for the codebooks, and the recognizer used to evaluate the codebooks as a speech representation. Chapter four describes the speech data base used in the study, and the manner in which it is processed, then discusses design and methodology of the two series of experiments, the first using codebooks comprised of vectors directly derived from synthesized sounds, and the second using codebooks trained from a synthesized seed codebook. Chapter five presents results for the series of experiments using synthesized codebooks, while chapter six presents results of the experiments using trained codebooks. Chapter seven compares the results with those found by other studies, discusses conclusions drawn from the experimental results, and suggests areas for further research.

Conventions

The following describes several conventions that are followed in this thesis.

The citation style is that recommended by the Acoustical Society of America, with periodical abbreviations being those published by the American Chemical Society (1974).

The most widely accepted notation for the sounds of speech is the international phonetic alphabet (IPA). Unfortunately, many of the IPA symbols are not available on standard keyboards and printers. Speech sounds are represented in this work by an IPA substitute, known as the Arpabet, after the Dept. of Defense Advanced Research Projects Agency, or ARPA, (and now known as DARPA) which funded a large speech recognition project in the 1970's. The version of the Arpabet used here consists entirely of upper case alphabetic characters, each symbol having an IPA equivalent. The Arpabet may be found in a number of sources; one is Parsons (1987).

Finally, algorithms in chapters 3 and 4 are presented as segments of C code (Kernighan and Ritchie, 1978), as they might appear if excerpted from a program.

REFERENCES

- Acoustical Society of America (1988). "Information for contributors to the Journal of the Acoustical Society of America," J. Acoust. Soc. Am. **83**, introductory pages.
- American Chemical Society (1974). Bibliographic Guide for Editors and Authors (Am. Chem. Soc., Washington, D. C.).
- Burton, D. K. (1987). "Text dependent speaker verification using vector quantization source coding," IEEE Trans. Acoust. Speech Signal Process., **ASSP-35**, 133-143.
- Bell, C. G., Fujisaki, H., Heinz, J. M., Stevens, K. N., and House, A. S. (1961). "Reduction of speech spectra by analysis by synthesis techniques," J. Acoust. Soc. Am. **33**, 1725-1736.
- Bonneau, H., and Gauvain, J. L. (1987). "Vector quantization for speaker adaptation," Proc. IEEE ICASSP 87, 1434-1437.
- Buzo, A., Gray, A. H., Gray, R. M., and Markel, J. D. (1980). "Speech coding based upon vector quantization," IEEE. Trans. Acoust. Speech Signal Process., **ASSP-28**, 562-574.
- Chomsky, N., and Halle, M. (1968). Sound Pattern of English (Harper and Row, New York).
- Compernelle, D. V. (1987). "Increased noise immunity in large vocabulary speech recognition with the aid of spectral subtraction," Proc. IEEE ICASSP 87, 1143-1146.
- Di Benedetto, M. G., and Lanaro, A. (1984). "How to avoid vowel normalization in identification of vowels in continuous speech," Proc. IEEE ICASSP 84, 42.4.1-42.4.4.

- Equitz, W. (1987). "Fast algorithms for vector quantization picture coding," Proc. IEEE ICASSP 87, 725-728; expanded paper submitted to IEEE Trans. Acoust. Speech Signal Process. (abstract published ASSP-35, 1361).
- Fant, G. (1973) "Descriptive analysis of the acoustic aspects of speech," in Speech Sounds and Features (MIT Press, Cambridge, MA), pp. 17-31.
- Furui, S. (1987). "A VQ based preprocessor using cepstral dynamic features for large vocabulary word recognition," Proc. IEEE ICASSP 87, 1127-1130.
- Gersho, A., and Cuperman, V. (1983). "Vector quantization: A pattern matching technique for speech coding," IEEE Commun. Magazine, December, 15-21.
- Golibersuch, R. J. (1983). "Automatic prediction of linear frequency warp for speech recognition," Proc. IEEE ICASSP 83, 769-772.
- Heinz, J. M., and Stevens, K. N. (1961). "On the properties of voiceless fricative consonants," J. Acoust. Soc. Am. 33, 589-596.
- IBM Speech Recognition Group. (1987). "Experiments with the Tangora 20,000 word speech recognizer," Proc. IEEE ICASSP87, 701-704.
- Kernighan, B. W. and Ritchie, D. M. (1978). The C Programming Language (Prentice-Hall, Englewood Cliffs, NJ).
- Klatt, D. H. (1974). "Word verification in a speech understanding system," in Speech Recognition, edited by D. R. Reddy (Academic Press, New York), pp. 321-341.

- Klatt, D. H. (1977). "Review of the ARPA Speech Understanding Project," J. Acoust. Soc. Am. **62**, 1345-1366; reprinted (1980) in Trends in Speech Recognition, edited by W. A. Lea (Prentice-Hall, Englewood Cliffs, NJ), pp. 249-271.
- Kopec, G. (1986). "Formant tracking using hidden Markov models and vector quantization," IEEE Trans. Acoust. Speech Signal Process. **ASSP-34**, 709-729.
- Lee, D. L., and Lochovsky, F. H. (1983). "Voice response systems," ACM Computing Surveys **15**, 351-374.
- Lieberman, A. M. (1970). "The grammars of speech and language," Cognitive psychology **1**, 301-323.
- Linde, Y., Buzo, A., and Gray, R. M. (1980). "An algorithm for vector quantizer design," IEEE Trans. Commun. **COM-28**, 84-95.
- Loeb, E. P., and Lyon, R. F. (1987). "Experiments in isolated digit recognition with a cochlear model," Proc. IEEE ICASSP 87, 1131-1134.
- Moore, R. K., Russell, M. J., and Tomlinson, M. J. (1983). "The discriminative network; a mechanism for focusing recognition in whole word pattern matching," Proc. IEEE ICASSP 83, 1041-1044.
- Olson, H. F., and Belar, H. (1956). "Phonetic typewriter," J. Acoust. Soc. Am. **28**, 1072-1081.
- Paliwal, K. K., and Rao, P. V. S. (1982). "Synthesis based recognition of continuous speech," J. Acoust. Soc. Am. **71**, 1016-1024.
- Parsons, T. (1987). Voice and Speech Processing (McGraw-Hill, New York).

- Reininger, H., and Wolf, D. (1985). "Speech and speaker independent codebook design in VQ coding schemes," Proc. IEEE ICASSP 85, 1700-1702.
- Roe, D. B. (1987). "Speech recognition with a noise adapting codebook," Proc. IEEE ICASSP 87, 1139-1142.
- Roucos, S., and Wilgus, A. M. (1985). "The waveform segment vocoder: A new approach for very low rate speech coding," Proc. IEEE ICASSP 85, 236-239.
- Shannon, C. E. (1948). "A mathematical model of communication," Bell Syst. Tech. J. 27, 379-423, 623-656.
- Silvertsen, E. (1961). "Segment inventories for speech synthesis," Language and Speech 4, 27-89.
- Soong, F. K., Rosenberg, A. E., and Juang, B. H. (1987). "A vector quantization approach to speaker recognition," AT&T Technical Journal 66, 14-26.
- Thosar, R. B., and Rao, P. V. S. (1976). "An approach towards a synthesis based speech recognition system," IEEE Trans. Acoust. Speech Signal Process. ASSP-24, 194-196.

CHAPTER 2

VECTOR QUANTIZATION IN SPEECH RECOGNITION

Even though the LBG algorithm wasn't published until 1980, a great deal of work has been done in vector quantization, and the literature is voluminous. This chapter will focus on vector quantization as it is used in speech recognition, where it has become the most popular data representation for large vocabulary systems (Bonneau and Gauvain, 1987).

Recognition Paradigms

The use of vector quantization in speech recognition falls naturally into two groups. These will be dubbed the recognizer based approach and the word based approach. The two methods have been combined in a hybrid approach in which a word based recognizer serves as a preprocessor to a conventional system (Pan et al., 1985; Furui, 1987). The 2 basic approaches, recognizer based and word based, are discussed in the following sections.

Recognizer-based methods

The more traditional of the two approaches uses vector quantization as a speech representation for a conventional speech recognizer -- a recognizer based on hidden Markov models (HMM) or on the use of dynamic time warping (DTW) to time align test utterance templates with those of the reference vocabulary. Figure 4 illustrates such a system.

Figure 4. Conventional speech recognizer with vector quantization.

Preprocessing includes low pass or band pass filtering of the analog speech signal, digitization by sampling the filtered signal, endpoint detection, and some form of data reduction, such as linear predictive coding (LPC). Input to the vector quantizer, then, is a

series of signal "frame" vectors, each representing 10 to 30 ms of speech. Each frame is vector quantized individually, and the resulting vectors are used in a conventional DTW or HMM scoring algorithm.

The reasons usually given for using vector quantization are reductions in both memory use and computation time, although the discriminative network feature is also sometimes cited (Ney *et al.*, 1986).

Hidden Markov models used with vector quantization are usually Bakis models using a fixed number of states for all words (Figure 5).

Figure 5. Bakis model.

The Bakis model allows only three transitions from each state: a loop transition back to the same state, a transition to the next state,

or a skip of the next state to its following state. A hidden Markov model is usually implemented using two matrices -- an **A** matrix of transition probabilities, where a_{ij} is the probability of moving from state i to state j , and a **B** matrix of observation probabilities, where b_{ij} is the probability of observing vector j in state i . In the Bakis model, only three diagonals of the **A** matrix are nonzero, corresponding to transitions from state i to states i , $i+1$, and $i+2$. A more general model allowing transitions to any succeeding state increases computation but does not improve performance (Rabiner, Levinson, and Sondhi, 1983). The use of a fixed number of states, typically five, to model all the words further simplifies the recognition.

Dynamic time warping is an optimization technique that finds the best path through a graph. In the case of speech recognition, the graph is the score matrix derived from matching every frame of the unknown, or test, utterance with every frame of the vocabulary, or reference, utterance. The algorithm is described in more detail in chapter 3. Dynamic time warping is extended to connected speech recognition by the one pass DTW algorithm (Bridle et al., 1983) and

the level building algorithm (Myers and Rabiner, 1981).

Both DTW and hidden Markov models have advantages in speech recognition. Hidden Markov models require much less computation during recognition but require massive amounts of training data (O'Shaughnessy, 1987). DTW, because it is a nonparametric method, requires much less training data. Rabiner, Levinson, and Sondhi (1983) compared a hidden Markov model recognizer against a DTW recognizer and found comparable performance for the two methods when used with a 64 vector codebook, although DTW without vector quantization performed slightly better.

Word based methods

The word-based approach to speech recognition was pioneered by Shore and Burton (1983). Their method is founded on the observation that the vector quantization process yields a distortion score that may be used as the basis for recognition. Figure 6 is a block diagram of a word based vector quantization recognizer.

Figure 6. Word based vq recognizer.

In its simplest form, a word based recognizer creates a separate codebook, from training utterances, for each word in the recognition vocabulary. During recognition, the test utterance is vector quantized using each vocabulary codebook; the match scores for the vocabulary words are the quantization distortions -- the vocabulary word associated with the codebook yielding the minimum distortion is chosen as the recognized utterance, making recognition a side effect of quantization.

The great advantage of the word based method is that it avoids time alignment of test and reference utterances; unfortunately, the temporal information is completely thrown away, since words that have the same speech sounds (such as "top" and "pot"), regardless of the order of the sounds, will generate identical codebooks -- this becomes critical as the vocabulary grows in size and complexity. One method which attempts to preserve the order of the speech sounds uses "multisection" codebooks (Burton *et al.*, 1985), wherein each vocabulary word is linearly broken into n sections, and a codebook is created for each section of each word. During recognition, the test utterance is broken into n sections, and each section is quantized by

the corresponding section codebook of each vocabulary word. The score for the test utterance against a particular reference utterance is the sum of the section distortion scores for the reference template's section codebooks. Another way to introduce temporal information is to combine each vector's quantization distortion with a temporal distance score based on the probability density function of the vector's time of occurrence in the utterance (Bergh *et al.*, 1985).

Research in Recognition Using Vector Quantization

Research issues in the use of vector quantization for speech recognition include codebook sizes and generation methods, linguistic units in codebooks, separation of energy from spectral shape, and codebook adaptation and speaker independence. The following sections examine each of these questions in greater detail.

Effect of vector quantization on recognition accuracy

For speaker dependent recognition, several studies show that vector quantization does not necessarily mean a significant degradation in performance. Sugamura *et al.* (1983) report a nominal decrease in accuracy for vector quantization using 256

vectors, while Landell et al. (1984) actually found a small increase in accuracy for codebooks of 256 and 512 vectors.

The results are less clear for speaker independent recognition. Rabiner, Levinson, and Sondhi (1983) lost little accuracy when using vector quantization with isolated digits, but, on a more demanding vocabulary (Rabiner et al., 1984b), degradation due to vector quantization was more significant.

Effect of codebook size on recognition

It might be logical to suppose that the number of phonemes of a language places a lower limit on the number of vectors needed to characterize it. The smallest viable codebook for English would then consist of about 40 vectors, since there are approximately 40 phonemes in English. There are several problems with this reasoning, however. As stated previously, it is impossible to find discrete slices of the acoustic signal which correspond to each phoneme; this tends to increase the number of vectors required, since each phoneme will be represented by multiple vectors according to context.

On the other hand, some phonemes are spectrally

indistinguishable from one another; the spectrum of /SH/ ("shush"), for example, is identical to that of /CH/ ("church") -- the two sounds are distinguished by temporal cues such as duration and abruptness of onset. Since a codebook is a spectral representation, such sounds should be represented by a common vector, thus reducing the number needed. The effect is even greater if acoustic input is filtered to a narrow bandwidth, as sounds such as /S/ ("cease"), /T/ ("tart"), /F/ ("fife"), and /TH/ ("thistle") all lose their spectral identity. Of course, the number of necessary vectors is further reduced if the language subset to be quantized does not cover all phonemes.

Given the difficulty of theoretically determining the optimal codebook size, the usual approach has been to empirically determine, by creating and testing codebooks of different sizes, the number of vectors needed to obtain adequate recognition accuracy.

In isolated word experiments, with vocabularies ranging from 641 Japanese city names to the English digits, small codebooks have performed surprisingly well relative to larger ones; Sugamura et al. (1983), for example, observed a decline of only 3.4% from 256

vectors to 16. Pieraccini and Billi (1983), using speaker independent codebooks with speaker dependent templates, found declines of less than 1 percentage point for both one sided and two sided vector quantization as codebook size decreased from 256 vectors to 64. Applebaum et al. (1987) used both dynamic time warping and hidden Markov models, and varied number of coefficients, distance measures, and method of speech analysis, as well as codebook size; across most conditions, the degradation in performance from 128 vectors to 8 was remarkably consistent.

It is difficult to compare continuous or connected speech experiments with isolated word tests, but it is interesting to note that at least one such study (Landell et al., 1984) reported a 5 percentage point increase in word error rate, using a 100 word vocabulary, as the codebook size decreased from 512 vectors to 16; these results are very similar to those reported in isolated word experiments.

Using word based recognition, Shore and Burton (1983) reported error rate dropping only 1.5 percentage points, for speaker dependent recognition on the TI 20 word database, as the number of vectors in each word based codebook was reduced from 64 to 4.

Similar results were reported for multisection codebooks (Burton et al., 1985), with the number of vectors in each section codebook ranging from 2 to 16.

Linguistic units

The research described so far takes an "information theoretic" approach to the speech recognition problem. Other work assumes a more "cognitive" perspective by attempting to relate codebook vectors to perceptual linguistic units. Mergel and Ney (1985) attempted to associate vector quantization labels with phonemes by using a phonetically guided clustering technique. The method was later extended to connected digit recognition using subword models with and without internal temporal structure (Ney et al., 1986). Codebook sizes ranged from 22 to 74 vectors, and a standard dynamic time warping algorithm was used. Rabiner and Soong (1985) investigated vowel recognition using codebooks with vectors clustered separately from each vowel.

One problem with the word based method, in its original form or with multisection codebooks, is that it is restricted to isolated word recognition. In order to extend the technique to continuous speech,

Kopec and Bush (1985) incorporated the word based approach into a phonetically based network recognizer. Each word in the recognition vocabulary was first manually segmented into rough phonemic (or allophonic) classes. A codebook was created for each segment (in some implementations, multiple codebooks were created for each segment); codebook sizes ranged from 1 to 16 vectors. During recognition, the test utterance was matched by traversing a network with branches corresponding to the phonetic classes. The test utterance was not segmented before recognition; the network was traversed using dynamic programming techniques. On an isolated digit recognition task using several different sets of utterances, speaker independent accuracy ranged from 96.5% to 99% for 16 vector codebooks, and from 93.8% to 95.4% for single vector codebooks. The system has recently been tested with connected digits (Bush and Kopec, 1987).

Codebook generation method

Most vector quantization systems have used the LBG algorithm, with the initial codebook generated either by using a set of representative data vectors or by splitting (Linde *et al.*, 1980). The

splitting method begins with a codebook of one vector which is the centroid of the entire data set. After every iteration, each vector is split into two new vectors which are a fixed distance apart. The process continues until the codebook contains the desired number of vectors. Rabiner, Sondhi, and Levinson (1983) used several variations of the splitting method and found that all led to codebooks having virtually the same average distortion. Glinski (1985) extended the splitting method to use an N-way split, where N is any power of 2. Landell et al. (1984) generated codebooks both by traditional clustering methods and by covering. Covering is a "follow the leader" algorithm in which the initial codebook is just the first data vector. Each succeeding data vector is matched against the existing codebook; if the best match exceeds a predetermined threshold, the data vector is added to the codebook as a new vector. Generation of codebooks by covering required significantly less computation and performed as well at all sizes as their clustered counterparts.

Use of energy in vector quantization

Codebooks trained by traditional methods can contain vectors which are spectrally identical, differing only in amplitude. For this reason, many speech data compression systems have separated the gain from the spectral shape, encoding it in a separate scalar codebook (Gray, 1984). It is not clear, however, how to incorporate this structure into the distance measure of a speech recognizer; some systems based on LPC or cepstral processing have avoided this problem by discarding the gain term, thus normalizing the spectral shape of codebook vectors (Bourland et al., 1984; Appelbaum et al., 1987).

Rabiner et al. (1984a; 1984b) incorporated energy into LPC codebooks as an additional vector coefficient and report that, because of significant correlation between spectral shape and energy, fewer codebook entries are required than when quantizing shape and energy separately; they also report improved recognition accuracy. Rabiner (1984) later extended the method to connected speech.

A more effective way of using energy may be to adapt the quantization to the changing energy level of the waveform. Such an

adaptive system can be very complex, but Gersho and Cuperman (1983) describe a simple method using energy and signal autocorrelation to assign speech frames to one of three classes, with each class having its own codebook. This thesis describes, in chapters 4 and 6, a similar method which uses 3 classes but only 2 codebooks; such adaptive techniques have apparently not been previously applied in speech recognition.

One sided vs. two sided quantization

Systems using two sided vector quantization are necessarily recognizer based. The advantage of two sided quantization is that distance scores can be precomputed for all pairs of codebook vectors, thus saving a great deal of run time computation. Landell et al. (1984) compared one sided with two sided quantization and found the error rate for two sided quantization about 1/3 higher for all codebook sizes (512 to 16 vectors). Pieraccini and Billi (1983) report similar results. Glinski (1985) found string error rates close for one and two sided quantization with 1024, 256, and 64 vector codebooks; two sided quantization was markedly worse for 16 vector codebooks.

Universal codebooks

A number of studies have investigated the effect on recognition performance of using a speaker independent, or universal, codebook rather than a speaker dependent codebook. Such a codebook would provide a speaker independent representation at the acoustic level and perform the data compression necessary for large vocabulary recognition.

As expected, error rates for universal codebooks are considerably higher than those for speaker dependent codebooks. Mergel and Ney (1985) report error rates for universal codebooks several times those for speaker dependent codebooks. Glinski's (1985) results are more hopeful, but the error rate for universal codebooks is still about twice that for speaker dependent codebooks.

The original word based experiments (Shore and Burton, 1983) showed speaker independent recognition accuracy, on the TI 20 word vocabulary, to be substantially lower than that for speaker dependent recognition, although the results were better for the digits subset. With multisection codebooks (Burton *et al.*, 1985), the difference on the digits was even less.

A partial step toward a speaker independent codebook is to use data from multiple speakers; the test speaker is part of the reference data base, but does not provide all the data for codebook generation. Landell et al. (1984) reported an increase of about 1/3 in the error rate for multiple speaker codebooks. Shore and Burton also performed such a test with their word based system, and reported a moderate increase in error rate, from 2% to 4.5%, for the TI 20 word vocabulary.

Adaptive codebooks

An emerging area of research is directed toward developing codebooks that adapt to speaker or environment. Roe (1987) achieved significant improvement in recognition under noisy conditions by adapting a codebook created under quiet conditions.

Bonneau and Gauvain (1987) created a phonetic codebook from manually segmented data spoken by one speaker, then adapted the codebook to new speakers by using a short training sequence. The adapted codebook performed significantly better than the original.

Adaptive techniques are clearly an option for large vocabulary recognition, when one speaker can't be expected to train every word.

The Tangora system (IBM Speech Recognition Group, 1987) maintains a vocabulary of 20,000 words with a 100 vector codebook that is adapted to the user during a 20 minute training session.

Summary

Since the publication of the LBG algorithm less than 10 years ago, vector quantization has become the most widely used speech representation in large vocabulary recognition systems. In a recognizer based system, quantization is simply a preprocessing step for a conventional speech recognizer. In a word based system, recognition is a side effect of quantization.

There are a number of research issues in vector quantization for speech recognition. The work presented in this thesis is most directly related to efforts aimed at developing a universal codebook. Unlike the known (statistical) ways of producing a universal codebook, however, the codebook generation method, described here in chapter 4, applies the speech science technique of analysis by synthesis to vector quantization, resulting in a codebook that is based on prevailing theories of speech production and perception.

The work presented here also has much in common with research

concerning codebook size, but, unlike other studies, the smaller codebooks used in this work are all subsets of the original synthesized codebook; the goal is to determine which of the synthesized vectors are necessary for coding the data set.

Finally, the minimal synthesized codebook is tested as a seed for conventional codebook training, in an adaptive vector quantizer similar to that described by Gersho and Cuperman (1983).

REFERENCES

- Applebaum, T. H., Hanson, B. A., and Wakita, H. (1987). "Weighted cepstral distance measures in vector quantization based speech recognizers," Proc. IEEE ICASSP 87, 1155-1158.
- Bergh, A. F., Soong, F. K., and Rabiner, L. R. (1985). "Incorporation of temporal structure into a vector quantization based preprocessor for speaker independent, isolated word recognition," AT&T Technical Journal **64**, 1047-1063.
- Bonneau, H., and Gauvain, J. L. (1987). "Vector quantization for speaker adaptation," Proc. IEEE ICASSP 87, 1434-1437.
- Bourland, H., Wellekens, C. J., and Ney, H. (1984). "Connected digit recognition using vector quantization," Proc. IEEE ICASSP 84, 26.10.1-26.10.4.
- Bridle, J. S., Brown, M. D., and Chamberlain, R. M. (1983). "Continuous connected word recognition using whole word templates," The Radio and Electronics Engineer, **53**, 167-175.
- Burton, D. K., Shore, J. E., and Buck, J. T. (1985). "Isolated-word speech recognition using multisection vector quantization codebooks," IEEE Trans. Acoust. Speech Signal Process., **ASSP-33**, 837-849.
- Bush, M. A., and Kopec, G. E. (1987). "Network based connected digit recognition," IEEE Trans. Acoust. Speech Signal Process., **ASSP-35**, 1401-1413.
- Furui, S. (1987). "A vq based preprocessor using cepstral dynamic features for large vocabulary word recognition," Proc. IEEE ICASSP 87, 1127-1130.

- Gersho, A., and Cuperman, V. (1983). "Vector quantization: A pattern matching technique for speech coding," *IEEE Commun. Magazine*, December, 15-21.
- Glinski, S. C. (1985). "On the use of vector quantization for connected digit recognition," *AT&T Technical Journal* **64**, 1033-1045.
- Gray, R. M. (1984). "Vector quantization," *IEEE ASSP Magazine* **1**(2), 4-29.
- IBM Speech Recognition Group. (1987). "Experiments with the Tangora 20,000 word speech recognizer," *Proc. IEEE ICASSP* **87**, 701-704.
- Kopec, G. E., and Bush, M. A. (1985). "Network based isolated digit recognition using vector quantization," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-33**, 850-867.
- Landell, B. P., Naylor, J. A., and Wohlford, R. E. (1984). "Effect of vector quantization on a continuous speech recognition system," *Proc. IEEE ICASSP* **84**, 26.11.1-26.11.4.
- Linde, Y., Buzo, A., and Gray, R. M. (1980). "An algorithm for vector quantizer design," *IEEE Trans. Commun.* **COM-28**, 84-95.
- Mergel, D., and Ney, H. (1985). "Phonetically guided clustering for isolated word recognition," *Proc. IEEE ICASSP* **85**, 854-857.
- Myers, C. S., and Rabiner, L. R. (1981). "A level building dynamic time warping algorithm for connected word recognition," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-29**, 284-297.
- Ney, H., Mergel, D., and Marcus, S. M. (1986). "On the automatic training of phonetic units for word recognition," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-34**, 209-213.
- O'Shaughnessy, D. (1987). Speech Communication: Human and Machine (Addison-Wesley, Reading, MA).

- Pan, K. C., Soong, F. K., and Rabiner, L. R. (1985). "A vector quantization based preprocessor for speaker independent isolated word recognition," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-33**, 546-560.
- Pieraccini, R., and Billi, R. (1983). "Experimental comparison among data compression techniques in isolated word recognition," *Proc. IEEE ICASSP 83*, 1025-1028.
- Rabiner, L. R. (1984). "On the application of energy contours to the recognition of connected word sequences," *AT&T Bell Laboratories Technical Journal* **63**, 1981-1995.
- Rabiner, L. R., and Soong, F. K. (1985). "Single frame vowel recognition using vector quantization with several distance measures," *AT&T Technical Journal* **64**, 2319-2330.
- Rabiner, L. R., Levinson, S. E., and Sondhi, M. M. (1983). "On the application of vector quantization and hidden Markov models to speaker independent, isolated word recognition," *Bell Syst. Tech. J.* **62**, 1075-1105.
- Rabiner, L. R., Sondhi, M. M., and Levinson, S. E. (1983). "Note on the Properties of a Vector Quantizer for LPC Coefficients," *Bell Syst. Tech. J.* **62**, 2603-2616.
- Rabiner, L. R., Sondhi, M. M., and Levinson, S. E. (1984a). "A vector quantizer incorporating both LPC shape and energy," *Proc. IEEE ICASSP 84*, 17.1.1 - 17.1.4.
- Rabiner, L. R., Sondhi, M. M., and Levinson, S. E. (1984b). "On the use of hidden Markov models for speaker independent recognition of isolated words from a medium size vocabulary," *AT&T Bell Laboratories Technical Journal* **63**, 627 - 641.
- Roe, D. B. (1987). "Speech recognition with a noise adapting codebook," *Proc. IEEE ICASSP 87*, 1139-1142.

- Shore, J. E., and Burton, D. K. (1983). "Discrete utterance speech recognition without time alignment," IEEE Trans. Inf. Theory, **IT-29**, 473-491.
- Sugamura, N., Shikano, K., and Furui, S. (1983). "Isolated word recognition using phoneme like templates," Proc. IEEE ICASSP 83, 723-726.

CHAPTER 3

SOFTWARE TOOLS

There were two major software tools used in the study. The first was the synthesizer used to create sounds that, after undergoing the same signal processing as the speech data base, formed the vectors in the synthesized codebooks. The second was the recognizer used to evaluate the codebooks as a speech representation. The following describes these two programs; the way in which they are used in the experiments is described in chapter 4.

The Synthesizer

SINSYN (SINe wave SYNthesizer), is a simple formant synthesizer based on the VOSIM (VOIce SIMulation) model (Kaegi and Tempelaars, 1978). Because the coretechs processor is sensitive to waveform, SINSYN uses exponentially decayed sine waves, rather than VOSIM's squared half sines, in order to produce more speechlike waveforms; several other changes were made for convenience.

SINSYN gets its control parameters from an input file. A sample rate of 8000 Hz is assumed. The parameters are:

- beginning formant frequency (in Hz);
- ending formant frequency;
- beginning fundamental frequency;
- ending fundamental frequency;
- maximum fundamental frequency deviation (%);
- beginning amplitude;
- ending amplitude;
- formant bandwidth (in Hz);
- number of formant sine waves per pitch period;
- periodic/aperiodic;
- vibrato rate in (Hz);
- number of pitch periods to generate.

Formant waveforms are concatenated by putting multiple sets of parameters in a single file. For example, the following parameters produce the first formant of an utterance going from /IY/ ("bead") to /AA/ ("hod");

```
250 250 125 125 2 1000 1000 50 2 1 5 200
250 750 125 125 2 1000 1000 50 6 1 5 10
750 750 125 125 2 1000 1000 50 6 1 5 200
```

This sound has three segments; all three start and end with a fundamental frequency of 125 Hz (the third and fourth parameters) and a maximum amplitude of 1000, so pitch and volume are constant. The first segment is /IY/, with a (first) formant of 250 Hz.

The third segment is /AA/, with a formant of 750 Hz. The second segment is a transition, with the formant changing from 250 Hz to 750 Hz over 10 pitch periods. The bandwidth of the formant does not change, staying at 50 Hz. This utterance also has a 2% vibrato at a rate of 5 Hz; these parameters are held over from the VOSIM model -- they would not be used in typical speech synthesis and, in fact, were not used in the recognition experiments. The parameters for the second formant of IY-AA:

```

2500 2500 125 125 2 200 200 100 20 1 5 200
2500 1125 125 125 2 2000 5000 50 20 1 5 10
1125 1125 125 125 2 5000 5000 50 9 1 5 200

```

The formants generated by SINSYN are exponentially damped sine waves. The damping factor is determined by the formant bandwidth under the relationship $\sigma = -\pi B$, where σ is the damping factor and B is the bandwidth (Fant, 1970). The waveforms are generated by indexing a sine table, as described by Chamberlin (1985). For each sample, the value from the sine table, ranging from 0 to 1.0, is multiplied by the maximum amplitude and by the decay factor. The decay factor is reset to 1.0 at the beginning of each pitch period and is multiplied by e^{σ} after every sample. Figures 7a and 7b

show the waveform and magnitude spectrum of the /IY/ generated by the parameters above.

In the stand-alone version of SINSYN, formants are produced separately and mixed by another program. This provides a great deal of flexibility, since parameters are controlled separately for each formant, any number of which may be mixed to build increasingly complex waveforms. For the recognition experiments, however, a simple version of SINSYN was embedded in the codebook generation program. The embedded version of SINSYN always produces a two formant waveform. The arguments to the embedded version are:

- fundamental frequency;
- maximum amplitude for the waveform;
- first formant relative weight;
- second formant relative weight;
- first formant bandwidth;
- second formant bandwidth;
- periodic/apperiodic.

The relative weights allow the amplitude contributions of each formant to be controlled. Note that the embedded synthesizer produces static waveforms -- frequencies and amplitudes are not allowed to vary. Also, there is no length parameter; the embedded synthesizer simply fills a fixed length buffer.

(a) waveform

(b) magnitude spectrum

Figure 7. Synthesized /IY/.

The Recognizer

The recognizer used in the experiments is based on the one pass connected recognition algorithm (Bridle *et al.*, 1983). This is the model recommended by Hieronymus and Majurski (1985) for speech recognition benchmarking and data base analysis. As described by Ney (1984), it is a dynamic time warping template matcher. Each word in the reference vocabulary is represented by one or more templates. The unknown pattern is matched against each reference template by dynamic programming. The label of the reference template with the lowest matching score against the unknown is returned as the recognized word. Although Ney describes a connected recognition algorithm, the model has been modified here to deal with isolated words; the only change necessary was simply to not allow transitions from one reference template to another.

Conceptually, the scoring process may be viewed in terms of a two dimensional matrix. In order to compute the score for the unknown against a given vocabulary template, the matrix is first filled in by computing the local distortion between each frame of the unknown, and each frame of the reference template. The local distortion

measure used in the experiments was the simple city block distance; to calculate the distance between frames x and y , having n vector coefficients:

```
for (i = score = 0; i < n; i++)
    score += abs(x[i] - y[i]);
```

Word match scores accumulate according to the endpoint and path constraints, and the frame transition rules.

The frame transition rules within templates are the same as Ney's; no frames may be skipped. Figure 8 illustrates the transition rules in terms of the score matrix.

Figure 8. Recognizer continuity constraints.

The point (i, j) can be reached only by a horizontal move from $(i-1, j)$, a vertical move from $(i, j-1)$, or a diagonal move from $(i-1, j-1)$.

There is no score penalty for nondiagonal moves, but consecutive such moves are not allowed. The general dynamic programming formula for the path score at point (i, j) , given the above transition rules, is:

$$D(i,j) = d(i,j) + \min\{D(i-1,j), D(i,j-1), D(i-1,j-1)\},$$

where $D(x,y)$ is the accumulated path score at point (x,y) , and $d(x,y)$ is the local distortion score at point (x,y) . Consecutive nondiagonal moves are prohibited in the recognizer by simply not considering points which were approached by nondiagonal moves. The algorithm for computing the score for point (i,j) :

```

if (nondiagonal_flag[i-1][j] == FALSE)
    x = D[i-1][j];
else x = INFINITY;

if (nondiagonal_flag[i][j-1] == FALSE)
    y = D[i][j-1];
else y = INFINITY;

z = D[i-1][j-1];

D[i][j] = min(x,y,z) + d(i,j);

```

where D is a two dimensional score matrix, $nondiagonal_flag$ is a two dimensional matrix whose values indicate whether each point was approached from a diagonal or nondiagonal move, and d is a function

which returns the distortion score for unknown frame i and reference template frame j . The implementation of the path constraints amounts to a heuristic, and the algorithm is no longer guaranteed to return the optimal path. Optimality can be maintained by keeping three scores for each point in the score matrix; this was done in an earlier version of the recognizer. Comparison between the earlier version and the current implementation showed no noticeable difference in recognition performance, and only minor differences in scores. Because the current version executes faster and uses less memory, the earlier implementation was discarded.

The combination of transition rules and endpoint and path constraints defines the limits of acceptable recognition paths. Since the recognizer forces endpoints to match exactly, acceptable paths must fall within a window defined by the intersections of lines of slope $1/2$ and 2 originating at each endpoint. This is illustrated by Figure 9; acceptable paths must fall within the shaded region. For isolated word recognition, it is unnecessary to calculate scores which fall outside the window, but because of its heritage as a connected word recognizer, the recognizer described here calculates

the entire score matrix. It does, however, employ a novel path pruning algorithm based on a ternary voicing decision.

Figure 9. Recognition window.

Recognizers have problems using voicing to direct the match because of the inherent difficulty in telling whether a particular segment of speech is voiced. One solution has been to simply use the log amplitude value as a weighted match coefficient (Rabiner, 1984). This has the advantage of using the energy envelopes of the words to

help recognition, but it is difficult to find the right weight, and such a parameter may increase the recognizer's sensitivity to noise. The solution adopted in these experiments is a ternary voicing decision, the values of which might be described as "definitely voiced," "definitely unvoiced," and "maybe voiced." These categories correspond to high, mid, and low range values of a property related to voicing. The actual property used is described in chapter 4 as part of the speech representation, but the method will work with any appropriate property, such as amplitude. The voicing value is used in computing the score at frame (i, j) as follows:

```

    if (p(unk[i]) == HIGH && p(ref[j]) == LOW ||
        p(unk[i]) == LOW && p(ref[j]) == HIGH)
        D(i, j) = INFINITY;
    else match(i, j);

```

where $p(\text{unk}[i])$ is the property value of the unknown at frame i , $p(\text{ref}[j])$ is the property value of the reference template at frame j , and match is a path growing function that incorporates the scoring algorithm given above. This says that if one frame is definitely voiced and the other is definitely unvoiced, then the path score at location (i, j) may be assigned a prohibitively high value, effectively pruning the path, without performing further computation.

REFERENCES

- Bridle, J. S., Brown, M. D., and Chamberlain, R. M. (1983). "Continuous connected word recognition using whole word templates," *The Radio and Electronics Engineer*, **53**, 167-175.
- Chamberlin, H. (1985). Musical Applications of Microprocessors (Hayden Book Co., Berkeley, CA).
- Fant, G. (1970). Acoustic Theory of Speech Production (Mouton, The Hague, The Netherlands).
- Hieronymus, J. L., and Majurski, W. J. (1985). "A reference speech recognition algorithm for benchmarking and speech data base analysis," *Proc. IEEE ICASSP 85*, 1573-1576.
- Kaegi, W. and Tempelaars, S. (1978). "VOSIM -- a new sound synthesis system," *J. Audio Eng. Soc.* **26**, 418-425.
- Ney, H. (1984). "The use of a one stage dynamic programming algorithm for connected word recognition," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-32**, 194-196.
- Rabiner, L. R. (1984). "On the application of energy contours to the recognition of connected word sequences," *AT&T Bell Laboratories Technical Journal* **63**, 1981-1995.

CHAPTER 4

DESIGN AND METHODOLOGY

The primary purpose of the experiments described in this thesis was to determine the viability of using synthesis as the basis of speech coding for recognition, and, in so doing, to find the smallest codebook viable for a given set of speech data. A secondary purpose of the experiments was to test the use of a synthesized codebook as a seed for training a conventional codebook. To these ends, the experiments were organized in 2 sets, or series. The first set of experiments dealt only with synthesized codebooks and attempted to fulfill the primary purpose, while the second set of experiments used codebooks trained from a synthesized seed in an attempt to fulfill the secondary purpose.

The following describes, first, the speech data and processing for the experiments, and, second, the design and methodology of the experiments.

Speech Data and Processing

Data base

The speech used in the experiments was taken from the Texas Instruments isolated word data base, which has been placed in the public domain by the National Bureau of Standards (NBS). The entire data base consists of the digits, 10 control words, and the alphabet; these experiments used only the digits and control words. This subset was first used to benchmark a number of speech recognizers in a well known study (Doddington and Schalk, 1981), and is referred to in the literature as the TI 20 word data base.

The speech was recorded from 16 adult speakers -- 8 males and 8 females; no attempt was made to control for dialect (National Bureau of Standards). Recordings were made in a low noise sound isolation booth, using an Electro-Voice RE-16 dynamic cardioid microphone, positioned approximately 2 inches from the speaker's mouth. Each speaker recorded material in 9 sessions, each separated by at least one day. During the first (enrollment or training) session, the speaker recorded 10 tokens of each word. These utterances are intended for training speaker dependent recognizers. During each of

the 8 succeeding sessions, each speaker recorded 2 tokens of each word. These 16 utterances are used to test recognition.

The data base is provided by the NBS in digital format on video cassette tape. Because these tapes are loaned to users and must be returned, the speech data was converted to analog and redigitized, using 16 bit samples at 8 kHz, by a DSC 200 audio converter (Digital Sound Corp., 1982). The resulting digital speech was saved on magnetic tape by the Scott Instruments research computer -- a Digital Equipment Corporation VAX 11/750.

Preprocessing

In order to form the representation for the experiments the speech was again converted to analog by the DSC 200, and passed via line input to a Scott Instruments VET3 speech terminal (Scott Instruments Corp., 1986). The VET3 filtered the analog input to telephone bandwidth (300 - 3300 Hz), and then sampled it at 8 kHz; sample size was 12 bits. Figure 10 is a diagram of the preprocessing system.

Figure 10. Speech preprocessing.

At this point, the signal was split into 2 streams. One, the "raw signal," was passed to the SI2010 without further processing. The other (the "clipped signal") was differentiated and infinitely peak clipped before being passed to the SI2010. Differentiation approximates the effect of passing a signal through a 6 decibel (dB) per octave high pass filter; the operation is often used in speech processing systems to counter the effect of the natural 6 dB / octave magnitude decline in the speech spectrum (Witten, 1982).

Infinite peak clipping removes amplitude information from a

signal by reducing it to a binary representation; if the amplitude of the sample is greater than zero, the sample is assigned a value of 1; otherwise, the sample is assigned a value of -1 (or 0 in some systems). Infinitely peak clipped speech is highly intelligible and it retains intelligibility in the presence of noise (Licklider et al., 1948). It was used in these experiments to provide amplitude normalization both within the speech data base and between the speech data base and the synthesized sounds. Figure 11 illustrates the effects of differentiation and infinite peak clipping on a sampled waveform; the sound is "oh" (from "no") spoken by a female speaker.

Differentiation and infinite peak clipping can be combined into a single operation by the following algorithm:

```

for (i = 0; !EOF; i++)
{ if (new_sample > old_sample)
    buffer[i] = 1;
  else buffer[i] = -1;
    old_sample = new_sample;
}

```

Both the raw and clipped signals were passed to the SI2010, in blocks of 80 samples (10 ms).

(a) sampled waveform

(b) differentiated

(c) infinitely peak clipped

Figure 11. Speech processing of vowel "oh"

Speech representation

The SI2010 performs an operation similar to the average magnitude difference function (Hess, 1983), but uses a sliding window centered around the sample at the focus of attention. The actual process, described in Scott, et al. (1987), is conceptually similar to autocorrelation with difference magnitudes substituted for multiplications.

For each input signal, raw and clipped, the SI2010 produced a set of 49 coefficients per 10 ms frame. All codebook matching and recognition was done using the first 8 coefficients of the clipped signal. The raw signal was used only to derive the "correlation envelope," which was used as a voicing measure both by the recognizer, described in chapter 3, and by the endpoint detector, described below. For each frame, the value of the correlation envelope is the result of subtracting the first coefficient from the maximum value among the first 16 coefficients. The first coefficient is calculated from adjacent samples; the second coefficient is calculated from lags of 2 samples; and so forth. During strongly periodic input, such as voiced speech, the first coefficient is expected

to be low, since neighboring samples are close together, while the well defined peaks and valleys of a voiced waveform should cause higher coefficients to have high values. The random nature of aperiodic input, such as frication or silence, should produce coefficients all having about the same value, with perhaps slightly higher values for the first few coefficients due to high frequencies. Under most conditions, then, the correlation envelope should roughly mirror the energy envelope, with low values during frication and silence, and high values during voicing. In addition, the correlation envelope should be more reliable in high noise, since it takes into account the signal autocorrelation as well as energy.

Endpoint determination

Utterance endpoints were automatically determined by the VET3. The VET3 endpoint routine uses a correlation envelope threshold to detect voicing. A valid utterance must begin with at least 60 ms of voicing; input may drop below the threshold for up to 300 ms after the minimum voicing duration has been satisfied. Utterance endpoints are set 150 ms before the onset and after the offset of voicing. Figure 12 illustrates the operation of the endpoint detection

algorithm on an example utterance.

Figure 12. Endpoint detection using correlation envelope.

Frames of speech falling within the endpoints were uploaded to the VAX 11/750 for further processing. There, they were added in groups of three to create 30 ms frames.

Experimental Design

Test design

Recognition tests were designed following the guidelines set forth by Pallet (1985) for benchmarking recognition systems. The recognizer was forced to make a choice -- there was no option to reject a poorly matching utterance. Test documentation included confusion matrices and recognition scores for each word at 5 levels of

accuracy, depending on whether the correct word was in the top 5 matches (level 5), the top 4 matches (level 4), and so forth. The confusion matrices and word scores were provided cumulatively for each experiment as well as for each speaker.

All recognition was speaker dependent. The templates were not pooled -- all of each speaker's training tokens for each word were used as templates by the recognizer. Since there were 10 training tokens for each word, 200 templates were matched against each test utterance. The first experiment used no vector quantization; test utterances were directly matched against the vocabulary templates in order to provide a baseline figure to compare against experiments using vector quantized speech. There were two series of further experiments; the first used synthesized codebooks, while the second used trained codebooks based on a synthesized seed derived from the first series.

Codebook generation

The original codebook was designed to cover a 2 dimensional space determined by the observed ranges of first and second speech formants; the same procedure has often been followed in creating

synthesis matrices for speech perception experiments (Delattre et al., 1952; Miller, 1953; Ainsworth and Millar, 1971). The sound matrix was organized in columns determined by first formant frequencies. There were 27 columns in the matrix, with the first formant ascending from 250 Hz to 3625 Hz in 125 Hz intervals. The first entry in each of the first 10 columns (250 - 1375 Hz) was the first formant only. Successive column entries were mixtures of the first formant with second formants at 250 Hz intervals. The first column, then, was 250 Hz, followed by 250 + 500 Hz, then 250 + 750 Hz, et cetera, through 250 + 3000 Hz. Because the second formant falls, in speech, as the first rises, the highest second formant was lowered by 125 Hz in each of the first 10 columns -- the highest second formant in the first column was 3000 Hz; the highest second formant in the second column was 2875 Hz. At the eleventh column, the falling second formant met the rising first formant; at this point the first formant was no longer entered unmixed into the column, and the eleventh through twenty-sixth columns had 2 entries, each a second formant mixed with the first formant. The last two columns had only 1 entry each, with the last being the mixture of 3625 Hz with 3875

Hz.

Figure 13 shows the F1 - F2 coverage of the 109 original codebook.

Figure 13. F1 - F2 coverage in 109 vector synthesized codebook.

It may be argued that the high frequency coverage of the codebook was unnecessary, since the second formant in adult speech

should not exceed 3000 Hz (Peterson and Barney, 1952). The purpose of the high frequency "tail" starting at the eleventh column was to accommodate unvoiced speech, such as fricatives and bursts, which are characterized by high frequencies.

The entries in the sound matrix were synthesized by the embedded sine wave synthesizer with a 70% weighting on the first formant and 30% on the second formant; bandwidth was 50 Hz for both formants. Synthesized sounds were differentiated, infinitely peak clipped, and passed to an embedded SI2010 emulation routine, where they were processed in frames of 240 samples (30 ms) to form one set of coefficients for each sound. These coefficient vectors made up the original codebook of 109 entries.

Experiments using synthesized codebooks

The first set of experiments tested recognition accuracy using increasingly smaller synthesized codebooks. All codebooks were descended from the 109 vector original. Sizes tested were 109, 64, 48, 32, 24, 18, 16, 14, and 12 vectors. Smaller codebooks were derived by choosing the vectors most used by larger codebooks in vector quantizing the training data across all speakers, the argument

being that the vectors which are used least are most likely to be extraneous (of course, this is not guaranteed to be true). Because of the minimal use of many of the vectors in the 109 vector codebook, codebooks of 64, 48, and 32 vectors were derived directly from it. Further codebooks were taken from the immediately larger size by taking the vectors that were most often matched by voiced input; the 24 vector codebook was derived from the 32 vector codebook, 18 from 24, and so forth. The sizes smaller than 32 vectors were not preplanned, but were chosen somewhat arbitrarily based on the vector use. Below 18 vectors, so many matches were displaced that vector elimination proceeded by twos.

Recognition used two sided vector quantization -- both the vocabulary templates and test utterances were vector quantized by choosing, for each frame of speech, the closest matching vector from the codebook. The codebook was matched using the city block distance measure, as described in chapter 3. In the subsequent recognition, codebook vectors were substituted for speech frame coefficients, with only the correlation envelope values being retained from the original speech frames for both vocabulary and test

utterances. Figure 14 is a "before and after" view of an utterance as vector quantized by the 109 vector codebook; the dashed curve is the correlation envelope, indicating voicing. Figure 15 is a system diagram for the synthesized codebook experiments.

(a) original

(b) quantized by 109 vector codebook

Figure 14. Vector representation of word "one."

Figure 15. System diagram -- synthesized codebook experiments.

Experiments using trained codebooks

Since the most significant drop in performance in the synthesized codebook experiments was between 16 and 14 vectors, the 16 vector codebook provided the seed for training codebooks in further experiments.

The purpose of these experiments was to test the use of a synthesized codebook as a seed for conventional codebook training. In order to provide a consistency check, the speech data base was split into 2 parts, with speakers 1 - 4 (males and females) in one part, and speakers 5 - 8 in the other; a separate codebook was trained for each set of speakers.

Training was split on correlation envelope value; all frames having a low correlation envelope value, and mapping to the same vector in the 16 vector synthesized codebook, were averaged together. High correlation envelope frames were averaged in the same way. Speech frames having a correlation envelope value between the two thresholds were not averaged into the trained codebook. This training procedure resulted, for each set of speakers, in a codebook of 31 vectors -- 15 low correlation envelope vectors

and 16 high correlation envelope vectors. Only the training utterances were used in creating the trained codebooks. Vector quantization was performed adaptively in all experiments -- low correlation envelope frames were matched against the codebook's low correlation envelope vectors; high correlation envelope frames were matched against the codebook's high correlation envelope vectors. Mid correlation envelope frames were matched against the entire codebook.

There were 5 experiments in this series; the first experiment used two sided vector quantization on the trained codebooks. The codebooks were used in a multi-speaker fashion -- test and training data for speakers 1 - 4 were vector quantized using the codebook trained over speakers 1 - 4; data for speakers 5 - 8 were vector quantized using the codebook trained over speakers 5 - 8.

If the synthesized codebook provided a particularly good seed, then the codebook resulting from a single pass of training should not be much improved by additional training. The second experiment attempted to test this by using codebooks recursively trained in a manner similar to the LBG algorithm (Linde et al., 1980). The

training data for each set of speakers was mapped to the codebooks used in the first experiment. Frames matching to the same vectors were averaged to create a new codebook; again, the training was split on correlation envelope, and mid correlation envelope frames were not used. The process was continued, for each data set, until fewer than 1,000 vectors changed match assignment on a training pass (there were approximately 35,000 vectors in each set of training data); for each codebook, this resulted in a total of 9 training passes. All other aspects of the experiment were the same as in the first experiment.

The third and fourth experiments were designed to determine the highest recognition accuracy that could be attained using the single pass trained codebooks. First, for the third experiment, one sided vector quantization was used -- the raw speech frames from the test data were matched against the vector quantized frames of the training data. The fourth experiment continued one sided vector quantization, but, in order to place less emphasis on signal coefficients that varied widely, used a modified distance measure which might be called the normalized city block distance. This is

similar to the normalized euclidean distance measure (Parsons, 1987), in that the score is normalized or weighted according to the standard deviation of each vector coefficient:

```

for (i = score = 0; i < n; i++)
{ temp = abs(x[i].value - y[i].value);
  temp -= x[i].sd;
  if (temp > 0)
    score += temp;
}

```

where x is a frame of the vector quantized training data, y is a frame of the test data, $x[i].value$ and $y[i].value$ are vector coefficients, and $x[i].sd$ is the standard deviation of vector coefficient $x[i]$. The frame score is normalized by subtracting, for each coefficient, the standard deviation of the trained vector coefficient from the absolute value of the difference between the two frame coefficients. If this results in a negative value, then it is simply not added to the frame score. The measure was first presented by Scott and Hardesty (1983), where heuristically derived "variances" were used instead of standard deviations.

Finally, the fifth experiment in the trained codebook series attempted to determine the speaker dependence of the codebooks by crossing the data sets -- speakers 1 - 4 used the codebook trained

over speakers 5 - 8, and vice versa.

Figure 16 is a system diagram for the trained codebook experiments; recall that the initial codebook is the 16 vector synthesized seed codebook.

Figure 16. System diagram -- trained codebook experiments.

REFERENCES

- Ainsworth, W. A., and Millar, J. B. (1971). "Methodology of experiments on the perception of synthesized vowels," *Language and Speech* **14**, 201-212.
- Delattre, P., Liberman, A. M., Cooper, F. S., and Gerstman, L. J. (1952). "An experimental study of the acoustic determinants of vowel color," *Word* **8**, 195-210.
- Digital Sound Corporation. (1982). DSC 200 User Manual.
- Doddington, G. R., and Schalk, T. B. (1981). "Speech recognition: Turning theory to practice," *IEEE Spectrum* **18**(9), 26-32.
- Hess, W. (1983). Pitch Determination of Speech Signals (Springer-Verlag, New York).
- Licklider, J. C. R., Bindra, D., and Pollack, I. (1948). "The intelligibility of rectangular speech waves," *Am. J. Psychol.* **61**, 1-20.
- Linde, Y., Buzo, A., and Gray, R. M. (1980). "An algorithm for vector quantizer design," *IEEE Trans. Commun.* **COM-28**, 84-95.
- Miller, R. L. (1953). "Auditory tests with synthetic vowels," *J. Acoust. Soc. Am.* **25**, 114-121.
- National Bureau of Standards, "The Texas Instruments isolated word data base - 46 word version - PCM/VCR format."
- Pallet, D. S. (1985). "Performance assessment of automatic speech recognizers," *J. Res. Nat. Bur. Stand.*, vol. 90, no. 5, Sept.-Oct., 371-387; reprinted (1986) in Electronic Speech Recognition, edited by G. Bristow (McGraw-Hill, New York), pp. 277-306.

- Parsons, T. (1987). Voice and Speech Processing, (McGraw-Hill, New York).
- Peterson, G. E., and Barney, H. L. (1952). "Control methods used in a study of the vowels," J. Acoust. Soc. Am. **24**, 175-184.
- Scott, B. L., Goodman, R. G., Newell, J. M., and Smith, L. A. (1987). "Method and apparatus for generating a signal transformation and the use thereof in signal processing," United States patent application, pending.
- Scott, B. L. and Hardesty, L. H. (1983). "Method and apparatus for speech recognition," United States Patent no. 4,383,135.
- Scott Instruments Corporation. (1986). Coretechs VET3 Speech Terminal User Manual.
- Witten, I. H. (1982). Principles of Computer Speech (Academic Press, New York).

CHAPTER 5

RESULTS OF EXPERIMENTS USING SYNTHESIZED CODEBOOKS

The purpose of the first series of experiments was to determine the smallest viable synthesized codebook for the given (TI 20) data base. To this end, there were 8 experiments using synthesized codebooks; with the baseline experiment (no vector quantization), there were a total of 9 experiments in the series. The results discussed in this chapter are for selected experiments -- confusion matrices and recognition accuracy by speaker are given for all experiments in appendices A and B, respectively.

Recognition Accuracy

The baseline experiment yielded a recognition accuracy, across all speakers, of 97.0%; one speaker, f1, scored 100%, while the lowest score was 90.9%, for f8. As expected, accuracy is lowered by vector quantization; recognition using the 109 vector codebook was 92.3%. Table 1 gives cumulative scores for baseline and synthesized codebook experiments; level 1 accuracy is "correct recognition" -- the

correct word was the recognizer's first choice; level 3 requires that the correct word be in the recognizer's top 3 choices, and level 5 accepts the correct word if it is in the recognizer's top 5 choices.

Table 1.--Recognition accuracy using synthesized codebooks

Experiment	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
Baseline (no vq)	5120	97.0	98.6	98.8
109 vectors	5120	92.3	98.4	98.7
64 vectors	5120	92.2	98.3	98.7
48 vectors	5120	92.2	98.5	98.7
32 vectors	5120	92.1	98.2	98.7
24 vectors	5120	91.3	98.2	98.7
18 vectors	5120	91.7	98.2	98.7
16 vectors	5120	91.8	98.1	98.7
14 vectors	5120	89.9	98.0	98.7
12 vectors	5120	89.7	98.0	98.7

Table 1 shows that the scores for correct recognition are about 5% below the baseline for codebook sizes ranging from 109 vectors to

16. For the 14 vector codebook, there is a further drop of 2%.

Allowing the recognizer to return the top 3 candidates enables it to perform about as well for any of the tested conditions, there being only .6 percentage point difference in the best score and the worst at level 3. It might be presumed that any utterance that does not score correctly at level 5 is an outlier in the database, perhaps because of misalignment by endpoint detection (at least one such case is known to have occurred).

Use of Codebook Vectors

The small effect of removing 77 vectors from the codebook, in going from 109 vectors to 64, 48, and, finally, to 32, is not surprising when the use of the codebook vectors is considered. There were 72,806 frames of speech in the training data; Figure 17 shows the top 64 vectors from the 109 vector codebook, determined by the number of times they were used in quantizing the training data. The use of the codebook vectors was extremely nonlinear; the least used vector in Figure 17 was matched only 28 times. In going from 109 vectors to 64 in the codebook, 215 frames, or .3%, were forced to find a new match. The 48 and 32 vector codebooks dispossessed 1,170 and

4,460 frames, respectively (1.6% and 6.1%).

Figure 17. Vectors sorted by use -- 109 vector codebook.

Of the 72,806 frames in the training data, 30,349 were unvoiced (correlation envelope less than 128), 40,032 were voiced (correlation envelope greater than 256), and 2,425 were unclassified (correlation envelope between 128 and 256). Figure 18 shows the 64 most frequently matched vectors from the 109 vector codebook broken into voiced and unvoiced classes, again ranked by number of times matched.

Figure 18. Vectors sorted by use, voiced vs. unvoiced -- 109 vector codebook.

Figure 18 shows that the unvoiced frames in the speech training data cluster more strongly than the voiced frames, with 2 vectors accounting for 15,394 frames, or just over half the unvoiced data. Such an effect was expected, since, given the bandwidth of the speech input, unvoiced sounds should tend to lose their spectral identity; this was the argument for basing further reductions in codebook size (below 32 vectors) on matching by voiced frames alone.

Figure 19 shows the use of codebook vectors for the 32 vector

codebook; again the vectors are ranked by number of times matched when quantizing the training data.

Figure 19. Vectors sorted by use -- 32 vector codebook.

Figure 19 indicates that the matching of voiced speech frames to the 32 vector codebook is almost linear, while unvoiced frames still cluster into 3 or 4 groups, with the top 2 vectors now accounting for over 52% of the unvoiced training data.

Figure 20 presents vector use information for the 16 vector codebook. The match of codebook vectors by voiced frames is

becoming more level, while vectors matched by unvoiced frames continue to cluster -- the top 2 vectors now account for almost 56% of the unvoiced data.

Figure 20. Vectors sorted by use -- 16 vector codebook.

Figure 21 shows vector use for the 14 vector codebook. The two most frequently used vectors for unvoiced frames are now nearly equal; together, they account for almost 64% of the unvoiced data.

Figure 21. Vectors sorted by use -- 14 vector codebook.

As can be seen from Table 1, the recognition performance was quite constant from 109 vectors to 16, with a break occurring between 16 and 14 vectors. The reason for the break is not apparent from comparison of Figure 21 with Figure 20.

Codebook Decimation

Figure 22 shows which vectors survived the cuts from 109 vectors to 32.

Figure 22. F1 - F2 coverage in 32 vector synthesized codebook.

Obviously, the high frequency vectors in the 109 vector codebook were not popular with the quantizer -- only two remain that have a component above 3 kHz. Most of the remaining vectors fall within what are usually considered the main parts of the speech frequency range, with F1 below 1,000 Hz, and F2 below 2,500 Hz.

Figure 23. F1 - F2 coverage in 16 vector synthesized codebook.

Figures 23 and 24 show the surviving vectors in the 16 and 14 vector codebooks, respectively. The two vectors lost in going from 16 vectors to 14 were the two highest F2 vectors.

Figure 24. F1 - F2 coverage in 14 vector synthesized codebook.

Quantization Distortion

Figure 25 shows the mean quantization distortion for each of the synthesized codebooks. The distortion scores are for the training data; scores for the test data are similar. As expected, the scores for unvoiced vectors were higher than those for voiced vectors; the mean scores for voiced vectors ranged from 125.5 for the 109 vector

codebook to 146.5 for the 12 vector codebook, while the range for unvoiced vectors was 208.1 to 221.4, with the two curves roughly parallelling one another. The mean scores for unclassified vectors were almost identical to the overall scores, thus indicating that the unclassified vectors are composed about equally of voiced and unvoiced vectors.

Figure 25. Mean distortion by codebook size.

Figure 25 exhibits a clear break between the mean distortion scores for the 16 vector and the 14 vector codebooks, corresponding

to the break in recognition accuracy. It may be argued, however, that if mean distortion is the only indicator of the codebook's adequacy, then there should be a drop in recognition accuracy between the 32 vector and the 16 vector codebooks similar to the drop between the 16 and 14 vector codebooks, since the difference in distortion scores is about the same.

Figure 26, showing the standard deviations of the codebook distortion scores, convincingly illustrates the deterioration of coding with the 14 vector codebook.

Figure 26. Variability of distortion scores by codebook size.

The variability of the distortion scores for unvoiced vectors declines slightly over the course of the experiments, while the variability of the scores for voiced vectors is almost constant from 109 vectors to 16. The jump in score variability for voiced vectors indicates a decline in the ability to discriminate between classes of these vectors for codebook sizes smaller than 16 vectors.

Effect of Vector Quantization on Recognition

Tables 2 and 3 are confusion matrices for the first two recognition experiments -- baseline and 109 vector codebook. A confusion matrix shows the patterns of recognition and misrecognition in a test of perception. In the tables presented here, rows represent words presented to the recognizer, while columns represent words returned by the recognizer. Table 2, for example, shows that, in the baseline experiment, "eight" was correctly recognized as "eight" 254 times and was recognized, incorrectly, twice as "help"; "enter" was correctly recognized 255 times and was recognized once as "two."

Table 2.--Confusion matrix for baseline experiment

	e i g h t	e n t r e	e r a v e r	f i o o v u r	f o o u r	g o o p e	h e l p e	n i o n	n o n	o n e	r e p e a t	r e u b o u t	s e i t h e n t	s e i t h e n t	s e i t h e n t	s e i t h e n t	t h e n t	t h e n t	y e s	z e r o	s c o r e (%)
eight	254	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	99.2
enter	.	255	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	99.6
erase	1	.	248	1	.	.	3	.	.	1	.	.	.	.	.	.	1	.	1	.	96.9
five	.	.	.	253	.	1	1	.	.	.	.	.	.	.	1	.	.	.	.	.	98.8
four	1	.	.	.	255	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	99.6
go	1	.	.	.	.	230	2	.	20	.	.	.	.	.	.	.	3	.	.	.	89.8
help	4	1	.	.	.	.	248	.	.	1	.	.	2	.	.	.	.	.	.	.	96.9
nine	1	1	.	.	.	.	.	250	.	2	.	.	.	.	.	.	2	.	.	.	97.7
no	.	.	.	.	.	25	.	.	230	1	.	.	.	.	.	.	.	.	.	.	89.8
one	.	.	.	.	.	.	.	.	.	253	.	.	1	.	1	.	.	.	.	1	98.8
repeat	3	.	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	.	.	.	.	.	.	.	.	.	255	1	.	.	.	.	.	.	.	99.6
seven	.	1	.	1	.	.	.	.	.	.	.	.	252	1	1	.	.	.	.	.	98.4
six	.	1	.	.	.	.	3	.	.	1	.	.	250	1	.	.	.	.	.	.	97.7
start	.	.	.	.	.	.	1	.	.	.	.	.	1	245	9	.	.	.	.	.	95.7
stop	2	.	.	.	.	.	10	.	.	1	1	.	.	2	240	.	.	.	.	.	93.8
three	3	1	.	.	.	.	.	.	.	.	.	.	1	.	.	.	248	3	.	.	96.9
two	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	254	.	.	99.2
yes	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	256	.	100.0
zero	5	1	.	.	.	4	.	.	2	.	1	.	.	.	.	.	5	.	238	.	93.0

Table 3.--Confusion matrix for 109 vector codebook

	e i g h t	e n t r e	e r a v e	f i o o v e r	f o o u e r	g o o e l p e	h e l p e	n e i o n e	n o n e	o n e	r e p e a t t	r e u b o u t	s e v e n	s i x a r t	s t o p e	t h r e e	t w o	y e s	z e r o	s c o r e (%)
eight	248	1	.	.	.	1	.	.	.	.	.	.	3	.	1	1	1	.	.	96.9
enter	1	253	.	.	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	98.8
erase	1	1	245	1	.	.	1	.	.	.	.	.	.	.	.	3	1	2	1	95.7
five	.	.	.	244	.	1	.	1	.	.	.	.	.	7	2	.	1	.	.	95.3
four	1	.	.	.	253	2	.	.	.	.	.	.	.	.	.	.	.	.	.	98.8
go	.	.	.	.	3	207	10	.	28	.	.	1	.	.	.	1	3	3	.	80.9
help	3	1	.	.	1	.	245	.	.	.	1	.	.	3	.	2	.	.	.	95.7
nine	1	1	3	2	.	.	.	221	4	14	.	1	5	.	.	.	2	2	.	86.3
no	.	.	.	.	.	46	.	.	202	3	.	.	.	.	.	1	.	2	.	78.9
one	.	.	.	3	1	.	.	5	.	242	.	1	3	.	.	.	.	.	1	94.5
repeat	3	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	1	1	1	2	.	1	.	2	.	246	1	.	.	1	.	.	.	96.1
seven	3	1	.	6	3	1	.	.	.	2	.	.	237	1	1	.	.	1	.	92.6
six	1	3	.	.	.	.	2	.	.	.	1	.	.	249	.	.	.	.	.	97.3
start	.	.	.	4	.	.	3	.	.	.	.	.	.	1	224	24	.	.	.	87.5
stop	2	.	.	.	1	.	15	.	.	.	.	2	.	.	9	227	.	.	.	88.7
three	1	1	1	1	1	.	.	.	.	.	.	.	2	1	1	.	248	7	.	93.8
two	5	.	.	.	.	4	7	.	1	.	.	.	5	.	.	.	.	233	.	91.0
yes	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	254	.	99.2
zero	5	1	.	.	2	15	2	.	15	.	1	3	2	.	.	.	9	.	201	78.5

As expected, recognition was slightly worse for most words when they were vector quantized. "Eight," for example, fell from 254 correct responses to 248, a drop in recognition rate from 99.2% to 96.9% (there were 256 trials for each word); "enter" fell from 99.6% (255 correct) to 98.8% (253 correct). Most words fell in this range -- about a 1% to 3% drop from the baseline condition when they were vector quantized. There were 5 words, however, that showed significantly greater drops in recognition due to vector quantization: go, nine, no, two, and zero.

The recognizer had trouble distinguishing between "go" and "no" even in the baseline condition; vector quantization only exacerbated the problem. "Nine" and "two" performed very well in the baseline experiment; with vector quantization by the 109 vector codebook, "nine" was often confused with "one," while "two" was confused with several other words. "Nine - one" distinctions are difficult for recognizers because of the nasality which may dominate both words. The "two" confusions are harder to explain, but may stem from the fact that "two" is not usually spoken with the articulators in the /UW/ position at the /T/ burst -- this causes a brief IH - EH - UW

transition, and, in fact, a pure /UW/ is often not reached. Three of the major confusions with "two" shared /EH/ as their first voiced sound ("eight," "help," and "seven"). "Two - eight" is a common recognizer problem, for the reason stated above and because of the acoustic similarity of /UW/ and /IY/ -- they share the same first formant. A weak /T/ burst looks, acoustically, much like /HH/, and, without the help of the third formant in distinguishing /L/, the recognizer could mistake "two" for "help." "Two - seven" is the hardest confusion to understand, but, over telephone bandwidth, /T/ and /S/ are spectrally alike and dynamic programming will wash out the temporal difference. Like /UW/, nasal sounds are dominated by a strong low first formant and, with the high frequencies filtered out, /V/ may easily be confused with /UW/. The other problem for "two" was "go," which is understandable because /OH/ and /UW/ are acoustically similar. At any rate, "two" performed better in the 109 vector experiment than any of the other words that had a large drop in performance from the baseline condition.

The biggest loser to vector quantization was "zero," with a recognition rate of 93% in the baseline experiment, but a rate of only

78.5% with the 109 vector codebook. As table 3 shows, the major confusions for "zero," with vector quantization, were "no" and "go"; this result, along with the "two - seven" confusions, indicate that the voiced fricatives, /V/ and /Z/, were not well represented in the synthesized codebook.

As can be seen from table 1, there was little difference in recognition accuracy for codebook sizes ranging from 109 vectors to 16, all these scores being within a range of 1%. The score for the 14 vector codebook, however, was 1.4% below this range, and was almost 2% lower than the score for the 16 vector codebook. Tables 4 and 5 are confusion matrices for the 16 vector and 14 vector codebook experiments.

The word showing the largest drop in recognition between the 16 and 14 vector codebooks is "three," which went from 241 correct responses (94.1%) to 217 (84.8%). Most of the new confusions for "three" were "two," indicating that the high frequency second formant of /IY/ is lost in the move from 16 vectors to 14; this is illustrated by Figure 27, which shows the original vector representation of a spoken "three" followed by the same utterance

vector quantized by both the 16 vector and 14 vector codebooks.

The utterance clearly loses a significant high frequency component when quantized by the 14 vector codebook.

Table 4.--Confusion matrix for 16 vector codebook

	e i g h t	e n t r e	e r a v e	f i o o	f o o	g o	h e l p	n e l n	n o	n e	r e p e a t	r u b o u t	s e v e n	s i x	s t a r t	s t o p	t h r e e	t w o	y e s	z e r o	s c o r e (%)
eight	247	1	.	.	.	2	.	.	.	.	.	.	2	.	.	1	3	.	.	96.5	
enter	.	250	1	.	1	.	.	.	.	.	.	.	.	1	.	.	1	2	.	97.7	
erase	1	1	248	1	.	.	.	1	.	.	1	.	.	.	.	1	1	1	.	96.9	
five	.	.	.	245	.	1	.	1	.	1	.	.	.	.	6	2	.	.	.	95.7	
four	1	.	.	.	253	2	.	.	.	.	.	.	.	.	.	.	.	.	.	98.8	
go	.	.	.	.	3	194	9	.	38	.	.	.	.	.	.	.	6	.	6	75.8	
help	3	1	.	.	1	1	242	.	.	1	.	.	2	2	3	.	.	.	.	94.5	
nine	.	1	5	1	.	.	1	220	1	16	.	1	7	.	.	2	1	.	.	85.9	
no	.	.	.	.	.	39	1	.	207	3	.	1	.	.	.	.	3	.	2	80.9	
one	.	.	.	5	.	.	.	6	.	240	.	2	1	.	.	.	.	1	1	93.8	
repeat	2	.	.	.	.	.	.	.	.	254	.	.	.	.	.	.	.	.	.	99.2	
rubout	.	2	1	4	.	1	.	.	2	.	242	.	.	1	1	1	.	1	.	94.5	
seven	2	2	.	2	2	.	.	.	1	.	.	245	.	.	.	1	1	.	.	95.7	
six	.	4	.	.	.	.	3	.	.	.	.	.	248	.	.	1	.	.	.	96.9	
start	1	.	.	1	1	.	2	.	.	.	.	.	.	210	41	.	.	.	.	82.0	
stop	2	.	.	.	.	.	12	.	1	.	2	.	.	10	229	.	.	.	.	89.5	
three	1	1	3	1	1	.	.	.	.	.	.	.	1	.	.	241	6	1	.	94.1	
two	6	.	.	.	.	2	4	.	2	.	.	4	.	1	.	1	234	.	2	91.4	
yes	3	.	2	.	.	.	.	.	.	.	.	.	.	1	.	.	.	250	.	97.7	
zero	4	2	.	.	.	13	2	.	15	.	1	.	.	.	.	1	19	.	199	77.7	

Table 5.--Confusion matrix for 14 vector codebook

	e i g h t	e n t r e	e r a v e	f i o v u	f o o r	g o o p	h e l p	n i n e	n o e	o n e	r e p e a t	r u b o u t	s e v e n	s i x	s t o p	s t o p	t h r e e	t w o	y e s	z e r o	s c o r e (%)
eight	244	.	.	.	.	.	1	.	.	.	.	.	5	1	1	1	3	.	.	.	95.3
enter	.	248	1	.	.	.	.	.	.	.	.	2	.	.	.	.	1	4	.	.	96.9
erase	1	1	240	1	.	.	.	1	.	.	1	.	.	.	.	6	.	5	.	.	93.8
five	.	.	2	240	.	.	.	2	.	1	.	.	1	.	6	3	.	.	1	.	93.8
four	2	.	.	.	250	1	1	.	.	.	.	.	.	.	1	1	.	.	.	.	97.7
go	.	.	.	.	3	193	7	.	37	.	.	.	2	.	.	.	.	8	.	6	75.4
help	3	1	.	.	.	.	233	.	.	.	1	.	1	2	2	10	.	1	1	1	91.0
nine	.	1	8	2	.	1	1	216	2	13	.	.	8	.	1	.	1	2	.	.	84.4
no	.	.	.	.	.	40	1	.	201	3	.	1	1	.	.	1	.	5	1	2	78.5
one	.	.	.	6	.	.	.	5	.	236	.	2	2	.	2	.	.	.	2	1	92.2
repeat	2	.	.	.	.	.	.	.	.	254	.	.	.	.	.	.	.	.	.	.	99.2
rubout	1	1	1	3	.	1	.	1	.	2	.	238	1	.	1	1	2	.	2	1	93.0
seven	2	2	.	2	1	.	.	.	.	2	.	.	242	.	1	1	.	1	2	.	94.5
six	8	4	.	.	.	.	1	.	.	.	1	.	.	241	1	.	.	.	.	.	94.1
start	2	.	.	2	.	.	1	.	.	.	.	.	2	216	30	.	.	3	.	.	84.4
stop	3	.	.	.	.	.	12	.	.	.	2	.	.	17	222	.	.	.	.	.	86.7
three	6	1	2	3	1	.	.	.	.	.	.	6	2	1	.	217	17	.	.	.	84.8
two	6	.	.	.	.	3	5	.	2	.	.	.	9	2	.	1	1	225	1	1	87.9
yes	4	.	1	.	.	.	.	.	.	.	.	.	1	3	.	.	.	247	.	.	96.5
zero	4	3	.	.	1	14	3	.	10	.	1	1	.	.	.	4	16	.	199	77.7

(a) original

(b) quantized by 16 vector codebook

(c) quantized by 14 vector codebook

Figure 27. Vector representation of word "three."

CHAPTER 6

RESULTS OF EXPERIMENTS USING TRAINED CODEBOOKS

Based on the results of the synthesized codebook experiments, the 16 vector synthesized codebook was chosen as the seed for training. As described in chapter 4, the trained codebooks were designed for adaptive vector quantization, with high correlation envelope (voiced) vectors averaged separately from those with low correlation envelope values (unvoiced). This process yielded trained codebooks of 31 vectors, as all 16 of the seed vectors were matched by voiced data while 15 of the seed vectors were matched by unvoiced data. There were two codebooks generated for each experiment, one for (male and female) speakers 1 through 4, and the other for speakers 5 through 8; because there was little difference in the results for the two speaker sets, the data will be combined except where noted. Again, confusion matrices and speaker recognition scores are given in appendices A and B.

Recognition Accuracy

There were 5 experiments in the trained codebook series. The first used codebooks trained by a single pass over the 16 vector seed, while the second used recursively trained codebooks. The third experiment in the series returned to the one pass codebook and used one sided vector quantization, while the fourth added a normalized distance measure. The final experiment continued one sided vector quantization with the normalized distance measure and crossed speakers with codebooks -- speakers 1 through 4 were vector quantized by the codebook trained on speakers 5 through 8, and vice versa. Table 6 gives the recognition scores for the trained codebook experiments, with the baseline score, from table 1, repeated for reference. As expected, training the codebooks improved the recognition performance; experiments using the normalized distance measure scored only about 2% below the baseline.

Table 6.--Recognition accuracy using trained codebooks

Experiment	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
Baseline (no vq)	5120	97.0	98.6	98.8
One pass training	5120	94.0	98.4	98.7
Recursive training	5120	94.0	98.5	98.8
One sided vq	5120	94.4	98.4	98.7
Normalized distance	5120	95.1	98.5	98.7
Cross spkr training	5120	94.9	98.5	98.7

Use of Vectors in Trained Codebooks

Figures 28 and 29 show the distributions of vector use in quantizing the training data with the one pass and recursively trained codebooks. There is no significant difference in the patterns of vector matching, although the recursively trained codebook for the speaker 5 - 8 set seems to be throwing away one of the voiced vectors.

Figure 28. Vectors sorted by use -- single pass trained codebooks.

Figure 29. Vectors sorted by use -- recursively trained codebooks.

Effect of Training on Quantization Distortion

Figures 30 and 31 show the number of changes of vector match assignment and the progression of quantization distortion over the recursive training sequence (since the first training pass created the codebooks, there were no match changes on the first pass). Both one pass codebooks yielded a mean distortion, over the training data, of approximately 88.5. After 9 passes of training, the mean distortions were 80.5 for the speaker 1 - 4 codebook, and about 82.1 for the speaker 5 - 8 codebook. Even though the 9 pass codebooks were used for the recognition experiment, the training process was continued to determine the lower limits of the mean distortions for the 2 codebooks. Over the additional training passes, the mean distortion score for both codebooks dropped by increasingly smaller amounts; after 29 passes the mean distortion for the speaker 1 - 4 codebook was 79.5, and the mean for the speaker 5 - 8 codebook was 81.5; after 49 passes the mean distortions were 79.4 and 81.4. For both codebooks, the mean distortion was higher for voiced vectors than for unvoiced; this effect is due to the more pronounced "shape" of voiced vectors. Silence and unvoiced speech generally have

relatively flat autocorrelation functions. Since the SI2010 performs an operation similar to autocorrelation, these vectors tend to be flat in the speech representation used in these experiments; this can be seen in Figure 32, which shows the utterance "one" in its original representation and as vector quantized by the speaker 1-4 trained codebook. Unvoiced vectors were poorly represented in the synthesized codebooks, so most of the improvement in training came from these vectors; the mean distortion for quantizing voiced test data dropped from 137.4, for the synthesized 16 vector codebook, to 98.0 (speakers 1 - 4) and 99.9 (speakers 5 - 8), for the one pass trained codebooks, while for unvoiced vectors the mean distortion dropped from 210.5 to 78.5 and 74.9. For the 9 pass codebooks, the mean distortion was 93.5 and 95.7 for voiced vectors, and 67.5 and 66.2 for unvoiced vectors.

Figure 30. Vector match changes during training.

Figure 31. Mean distortion during codebook training.

(a) original

(b) quantized by one pass trained codebook

Figure 32. Vector representation of word "one."

Effect of Training on Recognition

Table 7 is the confusion matrix for one pass trained codebooks.

Table 7.--Confusion matrix for one pass training

	e i g h t	e n t r e	e r a v e r	f i o o v e r	f o o u e r	g o o l n p e	h e i o n e	n o n e	n o n e	o n e	r e u e p e a u n t	r e u e p e a u n t	s e i t t h w e e c	s e i t t h w e e c	s e i t t h w e e c	t h w e e c	t h w e e c	y e e c	z e e c	s c o r e (%)	
eight	254	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	99.2
enter	2	251	1	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	.	.	98.0
erase	1	.	246	1	.	.	.	2	.	.	1	.	.	.	.	4	.	1	.	.	96.1
five	.	.	.	247	2	1	.	.	.	.	.	.	.	4	2	.	.	.	.	.	96.5
four	1	.	.	.	253	.	.	.	.	.	.	.	.	1	1	.	.	.	.	.	98.8
go	.	.	.	.	.	205	9	.	35	.	.	.	.	.	1	1	4	.	1	.	80.1
help	4	2	.	.	.	1	244	.	.	.	1	.	.	2	.	2	.	.	.	.	95.3
nine	1	1	.	.	.	.	.	243	.	10	.	.	1	.	.	.	.	.	.	.	94.9
no	.	.	.	.	.	34	2	.	215	1	.	1	1	.	.	1	.	1	.	.	84.0
one	.	.	.	.	.	.	.	2	.	251	.	.	2	.	.	.	.	.	.	1	98.0
repeat	4	.	1	.	.	.	.	.	.	.	251	.	.	.	.	.	.	.	.	.	98.0
rubout	.	.	.	.	.	1	.	1	.	.	1	248	.	.	.	4	.	.	.	1	96.9
seven	3	2	.	1	1	1	.	1	.	.	.	.	243	1	1	1	.	1	.	.	94.9
six	1	2	.	.	.	.	1	.	.	.	.	.	.	251	1	.	.	.	.	.	98.0
start	.	.	.	1	.	.	3	.	.	.	.	.	.	1	230	21	.	.	.	.	89.8
stop	2	2	.	.	.	.	8	.	.	.	.	.	.	.	12	232	.	.	.	.	90.6
three	5	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	236	11	3	.	92.2
two	4	.	.	.	.	1	2	.	.	.	.	.	1	.	.	.	1	247	.	.	96.5
yes	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	254	.	99.2
zero	3	2	.	.	1	10	2	.	8	.	1	2	.	.	.	1	13	.	213	83.2	

Comparison with table 4 (in chapter 5) shows that trained codebooks improved the recognition of most words as compared against the 16 vector seed codebook; the 2 words benefitting most were "nine" and "start," although several others were significantly improved. More importantly, recognition of confusable pairs of words, such as "go - no" and "nine - one", was improved for both, indicating that the representation is truly better, rather than simply favoring one speech sound over another. As indicated by table 6, recognition using recursively trained codebooks was little different from that using the one pass codebooks; the recursively trained codebooks did improve the recognition of "go," to 84.8%, but lost accuracy on "zero," to 80.5%.

One Sided Quantization and Cross Speaker Training

The last 3 experiments used one sided vector quantization, matching the test data directly against training data quantized by the one pass trained codebooks.

One sided quantization yielded surprisingly little advantage, with the overall recognition being improved only .4% (table 6). One sided quantization improved recognition for "go" and "stop" (85.9% and

94.1%, respectively), but lost ground on "no" and "zero" (79.7% and 79.7%). The normalized distance measure further improved "go," to 89.1%, and repaired the damage to "zero" (84%), but did not help "no."

Finally, recognition was affected very little by cross speaker training, in which the codebook trained on speakers 1 - 4 was used to quantize the training data for speakers 5 - 8, and vice versa; again one sided vector quantization was used and the normalized distance measure was used. Table 6 shows that overall recognition was down only .2% for cross speaker training; this indicates that training did not introduce significant speaker dependence (note that the codebooks are not claimed to be truly speaker independent -- this aspect is discussed in chapter 7).

CHAPTER 7

CONCLUSIONS

The work described in this thesis investigated the use of speech synthesis techniques in vector quantization by finding the smallest subset of a synthesized codebook viable for coding a particular data set, and then by using that subset as a seed for conventional codebook training. This chapter summarizes the results, placing them in the context of related work, and suggests avenues for further research.

Comparison With Related Work

Figure 33 shows results from 3 studies using vector quantization with dynamic time warping recognizers, along with results for synthesized codebooks. These results were selected as being the most directly comparable, based on baseline scores, the use of dynamic time warping recognizers, and codebook sizes (recall that the synthesized codebook size was actually 109 rather than 128). Still, there are a number of differences; Sugamura *et al.* (1983) used

one sided vector quantization ("SPLIT" method) with a speaker dependent vocabulary of 641 Japanese city names; Landell et al. (1984) used a 100 word continuous speech vocabulary guided by a finite state syntax (results are shown for 2 sided vector quantization with speaker dependent clustered codebooks); Applebaum et al. (1987) used speaker independent isolated digits + "oh" (results are shown for eighth order LPC analysis with cepstral distance measure and dynamic time warping). In spite of these differences, the results for all 3 are remarkably consistent.

Recognition using synthesized codebooks is comparable to that found in the other 3 studies for 16 vector codebooks, but synthesized codebooks perform poorly for larger sizes. Accuracy for codebooks trained from a synthesized seed (31 vectors) ranged from 94.0 % to 95.1%, which places these results squarely in the range spanned by the 32 vector codebooks shown in Figure 33.

Figure 33. Selected results from vector quantization studies.

The advantage found here for one sided over two sided vector quantization was small (94.4% vs. 94.0%), but in line with that found elsewhere; Landell *et al.* found an advantage for one sided quantization of 2.9% for 16 vectors (93.7% vs. 90.8%) and 1.3% (95.3% vs. 94.0%) for 32 vectors, but the difference was smaller for larger codebook sizes; Pieraccini and Billi (1983) found differences ranging from .3% to .8% with codebook sizes ranging from 64 to 256.

The improvement from the normalized distance measure was greater than that from one sided vector quantization; unfortunately,

no comparable studies were found.

The comparisons show that the 16 vector synthesized codebook performs about as well as codebooks of comparable size generated by conventional methods. The major advantage for the synthesized codebook is its flexibility -- if the recognizer's signal processing method changes, for example, the 16 sounds making up the final codebook can be easily reprocessed to form the new representation; codebooks generated by conventional methods would have to be regenerated from scratch using a computationally expensive procedure. This advantage holds even when using the synthesized codebook as a training seed if, as the results indicate, only a few training iterations are necessary to create a usable codebook. A further advantage of using the synthesized seed is that it seems more likely that such a codebook, based on speech theory, will converge during training to a globally optimal result than conventional seed codebooks.

Suggestions for Further Research

There are several aspects of the work presented here that may provide basis for further study.

Development of synthesized codebooks

The sounds that formed the synthesized codebooks were extremely simple mixes of sine waves representing 2 formant frequencies for each sound. A significant extension to this work would involve more sophisticated synthesis, perhaps by using sounds created by a frequency domain synthesizer (Witten, 1982). Another possibility is to research ways of adding the third formant to the representation; this should help with recognition of R and L sounds, but it would also add a third dimension to the sound matrix. Perhaps the size of the first generation codebook could be kept under control by generating "islands" of sound corresponding to the cardinal vowels, rather than filling out the entire matrix.

At any rate, the most important extension to the synthesized codebooks would be the addition of explicitly unvoiced sounds -- silence, frication, and aspiration. The much higher distortion scores, reported in chapter 5, for unvoiced as opposed to voiced speech frames indicates that unvoiced sounds are not represented well in the codebooks, and, in fact, this can be seen in comparing the original with the vector quantized utterance in Figure 14 (chapter 4). The

distributions of codebook vectors after decimation to 32, 16, and 14 vectors, Figures 22, 23, and 24 (chapter 5) show that the high frequency tail in the 109 vector codebook was not favored by the quantizer.

Development of trained codebooks

The trained codebooks used in these studies consisted of 15 unvoiced and 16 voiced vectors. Given the restricted bandwidth of the preprocessing (300 - 3300 Hz), it should not be possible to distinguish 15 different unvoiced sounds; recall that over 50% of the unvoiced speech frames were matched by just 2 vectors in the 109 vector synthesized codebook. Figures 28 and 29 (chapter 6), showing the vector match totals for the one pass and the recursively trained codebooks, suggest that some of the unvoiced codebook vectors are superfluous, and could be eliminated by a decimation process like that carried out on the synthesized codebooks.

The trained codebook experiments described here have introduced the concept of using a "soft" voicing decision to direct adaptive vector quantization. Given the spectral differences between voiced and unvoiced waveforms, and the difficulty of algorithmically

telling them apart (Hess, 1983), the method should be extended to use with voicing measures other than the one used here, as adaptive techniques become the norm in vector quantization.

Normalized city block distance measure

The use of vector coefficient standard deviations to normalize the frame distortion scores during recognition caused a definite, if somewhat small, improvement in accuracy. This measure should be extended to codebook matching, although it is unclear how the match should be implemented. Standard deviations for the coefficients do not exist until the codebook has been trained with at least one clustering iteration. If the normalization is used on ensuing training passes the normalization factors will change from pass to pass, which may affect the codebook's ability to converge to a stable point.

Furthermore, Applebaum et al. (1987) found that weighted measures that performed better with unquantized data reacted unpredictably with vector quantization systems.

Toward a universal codebook

The primary purpose of this work was to take a step in the

direction of a compact but universal speech representation. Figure 33 shows that the 16 vector synthesized codebook performed about as well as 16 vector codebooks in other systems, and the cross speaker codebooks, trained and tested on different speakers, performed comparably to other codebooks of similar size.

Universality cannot be claimed for these codebooks, however. Since codebook decimation was based on the use of vectors in quantizing training data, the only codebook completely free of speaker and, especially, vocabulary effects was the first generation synthesized codebook of 109 vectors.

The results presented in this thesis show that the process described here is suitable for development of a universal codebook, but some changes are necessary. First, in order to avoid vocabulary effects, codebook decimation must be based on a phonetically balanced set of words; such sets of words have long been used in performing "articulation tests" (Egan, 1948) for the purpose of evaluating speech communication systems.

Second, it is difficult to be entirely sure of eliminating speaker effects, but the method might be to use multiple sets of speakers,

each equally heterogeneous with respect to sex, age, and dialect (and even language, if the resulting codebook is expected to cross language boundaries) and to merge the codebooks from all the groups of speakers.

Finally, in order to determine precise phonetic strengths and weaknesses, candidate codebooks should be tested on recognition of minimal word pairs. Minimal pairs are words that differ from one another by a single phoneme, such as "tip - top" or "spud - spun." Such a test would be modelled after the Diagnostic Rhyme Test (Voiers, 1979), but, unlike the DRT, would test vowels as well as consonants. Tests could use the codebooks not only as a representation for automatic speech recognition, but also as a representation for speech synthesis, with listening tests on human subjects.

Summary

The experiments described here created a 109 vector codebook, using speech synthesis techniques, and found the smallest subset of that codebook viable for coding a particular speech data base. The resulting codebook was used as a seed and, after just one pass of

training, was able to achieve a recognition score within 2% of the baseline, which used no vector quantization.

The synthesized codebook developed here cannot be claimed to be speaker or vocabulary independent, but the same process, carried out over a phonetically balanced set of speech data, and with a suitable variety of speakers, would result in a minimal speaker and vocabulary independent codebook.

REFERENCES

- Applebaum, T. H., Hanson, B. A., and Wakita, H. (1987). "Weighted cepstral distance measures in vector quantization based speech recognizers," Proc. IEEE ICASSP 87, 1155-1158.
- Egan, J. P. (1948). "Articulation testing methods," Laryngoscope, 58, 955-991.
- Hess, W. (1983). Pitch Determination of Speech Signals (Springer-Verlag, New York).
- Landell, B. P., Naylor, J. A., and Wohlford, R. E. (1984). "Effect of vector quantization on a continuous speech recognition system," Proc. IEEE ICASSP 84, 26.11.1-26.11.4.
- Pieraccini, R., and Billi, R. (1983). "Experimental comparison among data compression techniques in isolated word recognition," Proc. IEEE ICASSP 83, 1025-1028.
- Sugamura, N., Shikano, K., and Furui, S. (1983). "Isolated word recognition using phoneme like templates," Proc. IEEE ICASSP 83, 723-726.
- Voiers, W. D. (1979). "Intelligibility testing at Dynastat: The diagnostic rhyme test," Dynastat Inc., Austin, TX.
- Witten, I. H. (1982). Principles of Computer Speech (Academic Press, New York).

APPENDIX A

RECOGNITION ACCURACY BY SPEAKER

Table 8.--Speaker scores for baseline experiment

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	100.0	100.0	100.0
f2	320	96.9	99.7	99.7
f3	320	95.0	97.8	98.1
f4	320	93.1	96.6	96.6
f5	320	97.5	99.7	99.7
f6	320	98.1	98.1	98.4
f7	320	98.4	99.4	100.0
f8	320	90.9	96.6	96.6
m1	320	95.3	97.8	98.8
m2	320	99.7	99.7	99.7
m3	320	96.9	98.4	98.8
m4	320	99.4	99.7	100.0
m5	320	95.6	97.8	97.8
m6	320	97.8	98.4	98.4
m7	320	99.1	99.1	99.1
m8	320	98.4	99.4	99.4

Table 9.--Speaker scores for 109 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	98.4	100.0	100.0
f2	320	92.5	99.7	99.7
f3	320	92.2	97.8	98.1
f4	320	89.7	96.3	96.6
f5	320	94.7	99.7	99.7
f6	320	93.8	98.1	98.1
f7	320	94.4	99.4	100.0
f8	320	88.4	96.3	96.6
m1	320	90.0	97.8	98.8
m2	320	96.3	99.7	99.7
m3	320	89.1	97.5	98.4
m4	320	95.0	99.4	100.0
m5	320	90.6	97.2	97.8
m6	320	91.3	98.1	98.1
m7	320	96.6	99.1	99.1
m8	320	83.4	97.8	99.1

Table 10.--Speaker scores for 64 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	98.1	100.0	100.0
f2	320	92.2	99.7	99.7
f3	320	91.9	97.8	98.1
f4	320	90.0	96.3	96.6
f5	320	94.4	99.7	99.7
f6	320	94.4	98.1	98.1
f7	320	95.0	99.4	100.0
f8	320	87.5	95.9	96.6
m1	320	90.3	97.8	98.8
m2	320	96.3	99.7	99.7
m3	320	89.7	97.2	98.4
m4	320	94.7	99.7	100.0
m5	320	90.0	97.2	97.8
m6	320	90.6	97.8	98.1
m7	320	96.9	99.1	99.1
m8	320	83.8	97.8	99.1

Table 11.--Speaker scores for 48 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	98.1	100.0	100.0
f2	320	92.2	99.7	99.7
f3	320	92.5	97.8	98.1
f4	320	89.7	96.3	96.6
f5	320	94.4	99.7	99.7
f6	320	94.4	97.8	98.1
f7	320	95.3	99.4	100.0
f8	320	87.5	95.6	96.6
m1	320	90.3	97.8	98.8
m2	320	96.6	99.7	99.7
m3	320	89.7	96.9	98.4
m4	320	94.4	99.7	100.0
m5	320	89.7	97.2	97.8
m6	320	90.0	97.8	98.1
m7	320	96.9	99.1	99.1
m8	320	83.8	97.8	99.1

Table 12.--Speaker scores for 32 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	98.4	100.0	100.0
f2	320	92.2	99.7	99.7
f3	320	91.6	97.8	98.1
f4	320	90.0	95.9	96.6
f5	320	93.8	99.4	99.7
f6	320	94.4	97.8	98.1
f7	320	95.0	99.4	100.0
f8	320	86.6	95.6	96.6
m1	320	89.7	97.5	98.1
m2	320	96.6	99.4	99.7
m3	320	91.6	97.5	98.1
m4	320	93.4	99.1	99.1
m5	320	89.4	96.9	97.8
m6	320	90.0	98.1	98.1
m7	320	97.2	99.1	99.1
m8	320	83.1	97.8	98.8

Table 13.--Speaker scores for 24 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	97.5	100.0	100.0
f2	320	91.3	99.7	99.7
f3	320	90.6	97.8	98.1
f4	320	89.1	95.6	96.3
f5	320	94.1	99.7	99.7
f6	320	93.1	97.8	98.1
f7	320	93.1	99.4	100.0
f8	320	86.3	95.6	96.6
m1	320	90.6	97.5	98.1
m2	320	96.3	99.7	99.7
m3	320	90.3	97.5	98.1
m4	320	93.4	99.4	100.0
m5	320	88.8	97.8	98.1
m6	320	89.7	97.8	98.1
m7	320	94.7	98.8	99.1
m8	320	82.8	97.2	98.8

Table 14.--Speaker scores for 18 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	99.1	100.0	100.0
f2	320	90.9	99.7	99.7
f3	320	89.1	97.8	98.1
f4	320	89.1	95.6	96.6
f5	320	94.1	99.7	99.7
f6	320	92.8	97.2	98.1
f7	320	92.8	99.4	100.0
f8	320	87.5	95.3	96.6
m1	320	90.0	97.8	98.1
m2	320	95.9	99.7	99.7
m3	320	88.8	97.2	97.8
m4	320	94.4	99.7	100.0
m5	320	90.6	97.8	98.1
m6	320	91.6	98.1	98.1
m7	320	96.3	98.8	99.1
m8	320	84.4	97.2	99.1

Table 15.--Speaker scores for 16 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	98.4	100.0	100.0
f2	320	91.9	99.7	99.7
f3	320	90.0	99.7	99.7
f4	320	89.4	95.9	96.3
f5	320	93.4	99.4	99.7
f6	320	92.5	97.2	98.1
f7	320	95.3	99.4	100.0
f8	320	87.2	95.9	96.6
m1	320	89.7	97.2	98.1
m2	320	96.3	99.7	99.7
m3	320	89.1	96.9	98.4
m4	320	93.1	99.7	100.0
m5	320	91.9	97.5	98.1
m6	320	90.3	97.8	98.1
m7	320	96.3	98.8	99.1
m8	320	83.4	97.2	99.1

Table 16.--Speaker scores for 14 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	96.3	100.0	100.0
f2	320	90.9	99.7	99.7
f3	320	89.4	97.8	98.1
f4	320	84.4	95.9	95.9
f5	320	95.0	99.7	99.7
f6	320	90.9	97.5	98.1
f7	320	93.1	99.4	100.0
f8	320	87.2	95.0	96.6
m1	320	89.4	97.5	98.4
m2	320	93.8	99.7	99.7
m3	320	84.7	97.2	98.1
m4	320	91.9	99.7	99.7
m5	320	87.2	96.9	97.8
m6	320	87.5	96.3	98.1
m7	320	94.1	98.8	99.1
m8	320	82.5	97.5	99.4

Table 17.--Speaker scores for 12 vector codebook

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	96.3	100.0	100.0
f2	320	89.4	99.7	99.7
f3	320	89.7	97.8	98.1
f4	320	87.2	95.0	96.3
f5	320	93.8	99.7	99.7
f6	320	92.2	97.5	98.1
f7	320	93.1	99.4	100.0
f8	320	87.5	95.0	96.6
m1	320	90.6	97.5	98.8
m2	320	94.7	99.7	99.7
m3	320	83.4	97.5	98.4
m4	320	90.6	99.7	99.7
m5	320	85.9	96.9	97.2
m6	320	87.2	97.2	98.1
m7	320	92.2	98.8	99.1
m8	320	81.6	97.5	99.4

Table 18.--Speaker scores for one pass training

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	98.1	100.0	100.0
f2	320	93.8	99.7	99.7
f3	320	91.9	97.5	98.1
f4	320	89.7	94.7	95.9
f5	320	95.9	99.7	99.7
f6	320	94.1	97.8	98.4
f7	320	96.3	99.1	100.0
f8	320	90.3	96.9	96.9
m1	320	89.7	97.2	97.8
m2	320	96.6	99.4	99.7
m3	320	95.0	98.4	98.4
m4	320	95.6	99.7	100.0
m5	320	92.8	97.5	97.5
m6	320	94.4	98.4	98.4
m7	320	97.5	99.1	99.1
m8	320	92.8	99.4	99.4

Table 19.--Speaker scores for recursive training

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	99.7	100.0	100.0
f2	320	93.8	99.7	99.7
f3	320	93.4	97.8	98.1
f4	320	90.3	95.9	96.6
f5	320	96.3	99.7	99.7
f6	320	94.1	98.1	98.4
f7	320	96.9	99.4	100.0
f8	320	88.1	96.9	96.9
m1	320	89.4	97.8	99.4
m2	320	94.1	99.4	99.7
m3	320	94.4	98.1	98.4
m4	320	94.1	99.7	100.0
m5	320	93.4	97.8	97.8
m6	320	94.7	98.1	98.4
m7	320	97.5	98.8	99.1
m8	320	93.4	98.8	99.4

Table 20.--Speaker scores for one sided vq

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	98.1	100.0	100.0
f2	320	94.7	99.7	99.7
f3	320	94.7	97.5	98.1
f4	320	90.6	95.6	95.9
f5	320	95.6	99.7	99.7
f6	320	93.4	98.1	98.1
f7	320	95.9	99.4	100.0
f8	320	89.4	96.6	96.6
m1	320	92.2	97.5	98.1
m2	320	97.2	99.7	99.7
m3	320	94.1	98.1	98.4
m4	320	96.3	99.7	100.0
m5	320	92.5	97.2	97.5
m6	320	95.0	98.1	98.4
m7	320	96.6	98.8	99.1
m8	320	93.4	99.1	99.4

Table 21.--Speaker scores for normalized distortion

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	97.8	100.0	100.0
f2	320	95.6	99.7	99.7
f3	320	94.4	97.5	98.1
f4	320	90.9	95.9	96.6
f5	320	97.5	99.7	99.7
f6	320	96.6	97.8	97.8
f7	320	96.6	97.8	97.8
f8	320	90.6	96.6	96.6
m1	320	93.1	98.4	98.4
m2	320	97.2	99.1	99.7
m3	320	93.8	98.4	98.8
m4	320	96.6	99.7	100.0
m5	320	92.8	96.9	97.5
m6	320	95.3	98.4	98.4
m7	320	98.1	99.1	99.1
m8	320	94.4	99.1	99.4

Table 22.--Speaker scores for cross speaker training

Speaker	No. Trials	Accuracy (%)		
		Level 1	Level 3	Level 5
f1	320	97.8	100.0	100.0
f2	320	94.7	99.7	99.7
f3	320	95.0	97.5	98.1
f4	320	91.9	96.3	96.6
f5	320	96.9	99.4	99.7
f6	320	95.9	97.8	97.8
f7	320	94.1	99.4	100.0
f8	320	90.3	96.6	96.6
m1	320	91.3	98.4	99.1
m2	320	98.1	99.7	99.7
m3	320	95.3	98.4	98.8
m4	320	97.5	99.7	100.0
m5	320	93.1	96.6	97.2
m6	320	96.3	98.4	98.4
m7	320	98.4	99.1	99.1
m8	320	92.2	99.1	99.1

APPENDIX B

CONFUSION MATRICES

Table 23.--Confusion matrix for baseline experiment

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e		e	o	e		r	p	e			o		r
	t	r	e							a	u	n		t		e					e
										t	t										(%)
eight	254	.	.	.	.	.	2	.	.	.	.	.	.	.	.	.	.	.	.	.	99.2
enter	.	255	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	99.6
erase	1	.	248	1	.	.	.	3	.	.	1	.	.	.	.	.	1	.	1	.	96.9
five	.	.	.	253	.	1	1	.	.	.	.	.	.	.	1	.	.	.	.	.	98.8
four	1	.	.	.	255	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	99.6
go	1	.	.	.	.	230	2	.	20	.	.	.	.	.	.	.	3	.	.	.	89.8
help	4	1	.	.	.	.	248	.	.	.	1	.	.	2	.	.	.	.	.	.	96.9
nine	1	1	.	.	.	.	.	250	.	2	.	.	.	.	.	.	2	.	.	.	97.7
no	.	.	.	.	.	25	.	.	230	1	.	.	.	.	.	.	.	.	.	.	89.8
one	.	.	.	.	.	.	.	.	.	253	.	.	1	.	1	.	.	.	.	1	98.8
repeat	3	.	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	.	.	.	.	.	.	.	.	.	255	1	.	.	.	.	.	.	.	99.6
seven	.	1	.	1	.	.	.	.	.	.	.	.	252	1	1	.	.	.	.	.	98.4
six	.	1	.	.	.	.	3	.	.	.	1	.	.	250	1	.	.	.	.	.	97.7
start	.	.	.	.	.	.	1	.	.	.	.	.	.	1	245	9	.	.	.	.	95.7
stop	2	.	.	.	.	.	10	.	.	.	1	1	.	.	2	240	.	.	.	.	93.8
three	3	1	.	.	.	.	.	.	.	.	.	.	1	.	.	.	248	3	.	.	96.9
two	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	254	.	.	99.2
yes	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	256	.	100.0
zero	5	1	.	.	.	4	.	.	2	.	1	.	.	.	.	.	.	5	.	238	93.0

Table 24.--Confusion matrix for 109 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e		e	o	e		r	p	e			o	r	e
	t	r	e							a	u	n		t		e					(%)
eight	248	1	.	.	.	.	1	.	.	.	.	.	.	3	.	1	1	1	.	.	96.9
enter	1	253	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	98.8
erase	1	1	245	1	.	.	.	1	.	.	.	.	.	.	.	.	3	1	2	1	95.7
five	.	.	.	244	.	1	.	1	.	.	.	.	.	.	7	2	.	1	.	.	95.3
four	1	.	.	.	253	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	98.8
go	.	.	.	.	3	207	10	.	28	.	.	1	.	.	.	1	3	3	.	3	80.9
help	3	1	.	.	1	.	245	.	.	.	1	.	.	3	.	2	.	.	.	.	95.7
nine	1	1	3	2	.	.	.	221	4	14	.	1	5	.	.	.	2	2	.	.	86.3
no	.	.	.	.	.	46	.	.	202	3	.	.	.	.	.	1	.	2	.	2	78.9
one	.	.	.	3	1	.	.	5	.	242	.	1	3	.	.	.	.	.	.	1	94.5
repeat	3	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	1	1	1	2	.	1	.	2	.	246	1	.	.	1	.	.	.	.	96.1
seven	3	1	.	6	3	1	.	.	.	2	.	.	237	1	1	.	.	1	.	.	92.6
six	1	3	.	.	.	.	2	.	.	.	1	.	.	249	.	.	.	.	.	.	97.3
start	.	.	.	4	.	.	3	.	.	.	.	.	.	1	224	24	.	.	.	.	87.5
stop	2	.	.	.	1	.	15	.	.	.	2	.	.	9	227	.	.	.	.	.	88.7
three	1	1	1	1	1	.	.	.	.	.	.	.	2	1	1	.	248	7	.	.	93.8
two	5	.	.	.	.	4	7	.	1	.	.	.	5	.	.	.	.	233	.	.	91.0
yes	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	254	.	99.2
zero	5	1	.	.	2	15	2	.	15	.	1	3	2	.	.	.	.	9	.	201	78.5

Table 25.--Confusion matrix for 64 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e		e	o	e		r	p	e			o	r	
	t	r	e							a	u	n		t		e					(%)
										t	t										
eight	248	1	.	.	.	.	1	.	.	.	.	.	.	3	.	1	1	1	.	.	96.9
enter	1	253	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	98.8
erase	1	1	245	1	.	.	.	1	.	.	.	.	.	.	.	.	3	1	2	1	95.7
five	.	.	.	245	.	1	.	1	.	.	.	.	.	.	7	1	.	1	.	.	95.7
four	1	.	.	.	253	1	.	.	.	1	.	.	.	.	.	.	.	.	.	.	98.8
go	.	.	.	.	3	206	11	1	28	.	.	.	.	.	.	1	.	3	.	3	80.5
help	3	1	.	.	1	.	242	.	.	.	1	.	.	3	1	4	.	.	.	.	94.5
nine	1	1	3	3	.	.	.	222	3	13	.	1	6	.	.	.	2	1	.	.	86.7
no	.	.	.	.	.	47	.	.	202	3	.	.	.	.	.	1	.	1	.	2	78.9
one	.	.	.	3	2	.	.	4	.	242	.	1	3	.	.	.	.	.	.	1	94.5
repeat	3	.	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	1	1	1	2	.	.	.	2	.	245	1	.	2	1	.	.	.	.	95.7
seven	3	1	.	6	3	1	.	.	.	2	.	.	237	1	1	.	.	1	.	.	92.6
six	1	3	.	.	.	.	2	.	.	.	1	.	.	249	.	.	.	.	.	.	97.3
start	.	.	.	4	.	.	2	.	.	.	.	.	.	1	222	27	.	.	.	.	86.7
stop	2	.	.	.	1	.	15	.	.	.	.	2	.	.	9	227	.	.	.	.	88.7
three	1	1	1	1	1	.	.	.	.	.	.	.	1	1	1	.	240	7	1	.	93.8
two	5	.	.	.	.	3	6	.	.	.	.	.	5	.	.	.	1	236	.	.	92.2
yes	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	254	.	99.2
zero	5	1	.	.	2	15	2	1	13	.	1	3	2	.	.	.	.	10	.	201	78.5

Table 26.--Confusion matrix for 48 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	248	1	.	.	.	.	1	.	.	.	.	.	.	3	.	1	1	1	.	.	96.9
enter	1	253	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	98.8
erase	1	1	245	1	.	.	.	1	.	.	.	.	.	.	.	.	3	1	2	1	95.7
five	.	.	.	245	.	1	.	1	.	.	.	.	.	.	7	1	.	1	.	.	95.7
four	1	.	.	.	253	1	.	.	.	1	.	.	.	.	.	.	.	.	.	.	98.8
go	.	.	.	.	3	205	11	1	28	.	.	.	.	.	.	1	.	4	.	3	80.1
help	3	1	.	.	.	.	240	.	.	.	1	.	.	3	2	5	.	1	.	.	93.8
nine	1	1	3	3	.	.	.	224	3	11	.	1	6	.	.	.	2	1	.	.	87.5
no	.	.	.	.	.	44	.	.	204	3	.	.	.	.	.	1	.	2	.	2	79.7
one	.	.	.	3	2	.	.	4	.	242	.	1	3	.	.	.	.	.	.	1	94.5
repeat	3	.	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	2	1	1	2	.	.	.	2	.	242	1	.	3	1	.	1	.	.	94.5
seven	3	1	.	6	3	1	.	.	.	2	.	.	237	1	1	.	.	1	.	.	92.6
six	1	3	.	.	.	.	2	.	.	.	1	.	.	249	.	.	.	.	.	.	97.3
start	.	.	.	3	.	.	2	.	.	.	.	.	.	1	223	27	.	.	.	.	87.1
stop	2	.	.	.	1	.	14	.	.	.	.	2	.	.	7	230	.	.	.	.	89.8
three	1	1	1	1	1	.	.	.	.	.	.	.	1	1	1	.	239	8	1	.	93.4
two	5	.	.	.	.	3	7	.	.	.	.	.	5	.	.	.	1	235	.	.	91.8
yes	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	254	.	99.2
zero	5	1	.	.	2	14	2	1	13	.	1	4	2	.	.	.	.	11	.	200	78.1

Table 27.--Confusion matrix for 32 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e		e	o	e		r	p	e			o		r
	t	r	e							a	u	n		t		e					e
										t	t										(%)
eight	245	1	.	.	.	.	3	.	.	.	.	.	.	4	.	1	1	1	.	.	95.7
enter	1	253	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	98.8
erase	1	1	248	1	.	.	.	1	.	.	1	.	.	.	.	.	1	1	1	.	96.9
five	.	.	.	246	.	1	.	.	.	.	.	.	.	.	7	1	.	1	.	.	96.1
four	1	.	.	.	252	2	1	.	.	.	.	.	.	.	.	.	.	.	.	.	98.4
go	.	.	.	.	3	205	10	1	29	.	.	1	.	.	.	1	.	4	.	2	80.1
help	3	1	.	.	.	.	239	.	.	.	1	.	.	3	2	6	.	1	.	.	93.4
nine	1	1	3	3	.	.	.	222	2	14	.	1	5	.	.	.	3	1	.	.	86.7
no	.	.	.	.	1	44	.	.	203	2	.	1	.	.	.	1	.	2	.	2	79.3
one	.	.	.	3	2	.	.	3	.	243	.	1	3	.	.	.	.	.	.	1	94.9
repeat	3	.	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	2	3	1	2	.	1	.	2	.	240	1	.	2	1	.	1	.	.	93.8
seven	2	1	.	4	2	2	.	.	.	3	.	.	240	.	1	.	1	.	.	.	93.8
six	1	3	.	.	.	.	2	.	.	.	1	.	.	249	.	.	.	.	.	.	97.3
start	.	.	.	4	.	.	3	.	.	.	.	.	.	1	220	28	.	.	.	.	85.9
stop	2	.	.	.	1	.	14	.	.	.	.	2	.	.	7	230	.	.	.	.	89.8
three	1	1	1	1	1	.	.	.	.	.	1	.	3	1	1	.	236	8	1	.	92.2
two	5	.	.	.	.	3	5	.	.	.	.	.	5	.	.	.	.	238	.	.	93.0
yes	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	254	.	99.2
zero	5	1	.	.	3	15	2	1	14	.	1	3	2	.	.	.	.	12	.	197	77.0

Table 28.--Confusion matrix for 24 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	246	1	.	.	.	.	2	.	.	.	.	.	.	3	.	1	1	2	.	.	96.1
enter	.	252	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.	1	1	.	98.4
erase	1	1	246	1	.	.	.	.	.	.	1	.	.	.	.	.	5	1	.	.	96.1
five	.	.	.	246	.	1	.	.	.	.	.	.	.	.	7	2	.	.	.	.	96.1
four	1	.	.	.	253	1	.	.	.	.	.	.	.	.	.	1	.	.	.	.	98.8
go	.	.	.	.	2	196	10	1	36	.	.	1	.	.	.	1	.	5	.	4	76.6
help	3	1	.	.	.	.	240	.	.	.	1	.	.	3	3	4	.	1	.	.	93.8
nine	1	1	3	2	.	.	.	218	2	19	.	.	5	.	.	.	4	1	.	.	85.2
no	.	.	.	.	.	56	.	.	191	3	.	1	.	.	.	1	.	3	.	1	74.6
one	.	.	.	6	1	.	.	4	.	240	.	1	3	.	.	.	.	.	.	1	93.8
repeat	2	.	.	.	.	.	.	.	.	.	254	.	.	.	.	.	.	.	.	.	99.2
rubout	.	1	2	5	.	1	.	.	2	1	.	235	1	.	3	3	.	.	1	1	91.8
seven	2	3	.	3	2	1	.	.	.	2	.	.	239	.	1	.	1	2	.	.	93.4
six	1	2	.	.	.	.	2	.	.	.	1	.	.	249	.	.	1	.	.	.	97.3
start	.	.	.	2	.	.	3	.	.	.	.	.	.	1	218	32	.	.	.	.	85.2
stop	2	.	.	.	.	.	14	.	.	.	.	2	.	.	9	229	.	.	.	.	89.5
three	1	1	1	1	1	.	.	.	.	.	.	.	2	1	.	.	238	9	1	.	93.0
two	5	.	.	.	.	2	5	.	.	.	.	.	6	.	.	.	1	236	.	1	92.2
yes	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	254	.	99.2
zero	4	2	.	.	2	12	3	1	19	.	1	1	3	.	.	.	.	11	.	197	77.0

Table 29.--Confusion matrix for 18 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	247	1	.	.	.	.	1	.	.	.	.	.	.	3	.	1	1	2	.	.	96.5
enter	.	253	.	.	1	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	98.8
erase	1	1	246	1	.	.	.	1	.	.	.	.	.	.	.	.	3	1	1	1	96.1
five	.	.	.	244	1	1	.	1	.	.	.	.	.	.	7	2	.	.	.	.	95.3
four	1	.	.	.	252	1	.	.	.	1	.	.	.	.	1	.	.	.	.	.	98.4
go	.	.	.	.	2	193	9	.	37	.	.	1	.	.	.	.	6	.	8	.	75.4
help	3	1	.	.	1	.	243	.	.	.	1	.	.	2	2	2	.	1	.	.	94.9
nine	.	1	3	2	.	.	1	220	2	16	.	.	7	.	.	.	2	2	.	.	85.9
no	.	.	.	.	.	47	.	.	201	2	.	.	.	.	.	2	.	2	.	2	78.5
one	.	.	.	3	.	.	.	3	.	245	.	1	2	.	.	1	.	.	.	1	95.7
repeat	2	.	.	.	.	.	.	.	.	.	254	.	.	.	.	.	.	.	.	.	99.2
rubout	.	2	1	4	.	.	.	.	.	2	.	242	.	.	.	2	.	.	2	1	94.5
seven	2	2	.	2	2	.	.	.	.	2	.	.	241	2	.	.	1	2	.	.	94.1
six	.	4	.	.	.	.	4	.	.	.	.	.	.	247	.	.	1	.	.	.	96.5
start	.	.	.	3	1	.	2	.	.	1	.	.	.	1	215	33	.	.	.	.	84.0
stop	2	.	.	.	.	.	12	.	.	.	.	2	.	.	7	233	.	.	.	.	91.0
three	1	1	3	1	1	.	.	.	.	.	.	.	.	1	.	.	236	11	1	.	92.2
two	4	.	.	.	.	2	3	.	2	.	.	.	3	.	1	.	1	240	.	.	93.8
yes	2	1	3	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	249	.	97.3
zero	4	2	.	.	1	12	2	1	22	.	.	.	.	.	.	.	.	18	1	194	75.8

Table 30.--Confusion matrix for 16 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	247	1	.	.	.	.	2	.	.	.	.	.	.	2	.	.	1	3	.	.	96.5
enter	.	250	1	.	1	.	.	.	.	.	.	.	.	.	1	.	.	1	2	.	97.7
erase	1	1	248	1	.	.	.	1	.	.	1	.	.	.	.	.	1	1	1	.	96.9
five	.	.	.	245	.	1	.	1	.	1	.	.	.	.	6	2	.	.	.	.	95.7
four	1	.	.	.	253	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	98.8
go	.	.	.	.	3	194	9	.	38	.	.	.	.	.	.	.	.	6	.	6	75.8
help	3	1	.	.	1	1	242	.	.	.	1	.	.	2	2	3	.	.	.	.	94.5
nine	.	1	5	1	.	.	1	220	1	16	.	1	7	.	.	.	2	1	.	.	85.9
no	.	.	.	.	.	39	1	.	207	3	.	1	.	.	.	.	.	3	.	2	80.9
one	.	.	.	5	.	.	.	6	.	240	.	2	1	.	.	.	.	.	1	1	93.8
repeat	2	.	.	.	.	.	.	.	.	.	254	.	.	.	.	.	.	.	.	.	99.2
rubout	.	2	1	4	.	1	.	.	.	2	.	242	.	.	1	1	1	.	1	.	94.5
seven	2	2	.	2	2	.	.	.	.	1	.	.	245	.	.	.	1	1	.	.	95.7
six	.	4	.	.	.	.	3	.	.	.	.	.	.	248	.	.	1	.	.	.	96.9
start	1	.	.	1	1	.	2	.	.	.	.	.	.	.	210	41	.	.	.	.	82.0
stop	2	.	.	.	.	.	12	.	1	.	.	2	.	.	10	229	.	.	.	.	89.5
three	1	1	3	1	1	.	.	.	.	.	.	.	.	1	.	.	241	6	1	.	94.1
two	6	.	.	.	.	2	4	.	2	.	.	.	4	.	1	.	1	234	.	2	91.4
yes	3	.	2	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	250	.	97.7
zero	4	2	.	.	.	13	2	.	15	.	.	1	.	.	.	.	1	19	.	199	77.7

Table 31.--Confusion matrix for 14 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	244	.	.	.	.	.	1	.	.	.	.	.	.	5	1	1	1	3	.	.	95.3
enter	.	248	1	.	.	.	.	.	.	.	.	.	2	.	.	.	.	1	4	.	96.9
erase	1	1	240	1	.	.	.	1	.	.	1	.	.	.	.	.	6	.	5	.	93.8
five	.	.	2	240	.	.	.	2	.	1	.	.	1	.	6	3	.	.	1	.	93.8
four	2	.	.	.	250	1	1	.	.	.	.	.	.	.	1	1	.	.	.	.	97.7
go	.	.	.	.	3	193	7	.	37	.	.	.	2	.	.	.	.	8	.	6	75.4
help	3	1	.	.	.	.	233	.	.	.	1	.	1	2	2	10	.	1	1	1	91.0
nine	.	1	8	2	.	1	1	216	2	13	.	.	8	.	1	.	1	2	.	.	84.4
no	.	.	.	.	.	40	1	.	201	3	.	1	1	.	.	1	.	5	1	2	78.5
one	.	.	.	6	.	.	.	5	.	236	.	2	2	.	2	.	.	.	2	1	92.2
repeat	2	.	.	.	.	.	.	.	.	.	254	.	.	.	.	.	.	.	.	.	99.2
rubout	1	1	1	3	.	1	.	1	.	2	.	238	1	.	1	1	2	.	2	1	93.0
seven	2	2	.	2	1	.	.	.	.	2	.	.	242	.	1	1	.	1	2	.	94.5
six	8	4	.	.	.	.	1	.	.	.	1	.	.	241	1	.	.	.	.	.	94.1
start	2	.	.	2	.	.	1	.	.	.	.	.	.	2	216	30	.	.	3	.	84.4
stop	3	.	.	.	.	.	12	.	.	.	.	2	.	.	17	222	.	.	.	.	86.7
three	6	1	2	3	1	.	.	.	.	.	.	.	6	2	1	.	217	17	.	.	84.8
two	6	.	.	.	.	3	5	.	2	.	.	.	9	2	.	1	1	225	1	1	87.9
yes	4	.	1	.	.	.	.	.	.	.	.	.	.	1	3	.	.	.	247	.	96.5
zero	4	3	.	.	1	14	3	.	10	.	.	1	1	.	.	.	4	16	.	199	77.7

Table 32.--Confusion matrix for 12 vector codebook

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	244	1	.	.	.	.	1	.	.	.	.	.	.	4	1	.	3	2	.	.	95.3
enter	.	248	1	.	.	.	.	.	.	.	.	.	2	.	.	.	.	1	4	.	96.9
erase	1	1	235	2	.	.	.	1	.	.	1	.	.	.	.	.	9	.	6	.	91.8
five	.	.	1	242	.	.	.	2	.	1	.	.	1	.	5	3	.	.	1	.	94.5
four	2	.	.	.	250	1	.	.	.	.	.	.	.	.	1	1	.	1	.	.	97.7
go	.	.	.	.	3	194	9	1	38	1	.	.	.	.	.	.	1	6	.	3	75.8
help	3	1	.	.	.	.	232	.	.	.	1	.	1	2	2	11	.	1	1	1	90.6
nine	.	1	10	2	.	1	1	214	3	12	.	.	7	.	1	.	2	2	.	.	83.6
no	.	.	.	.	.	47	1	.	194	3	.	1	1	.	.	1	.	5	1	2	75.8
one	.	.	2	4	1	.	.	6	.	237	.	2	1	.	2	.	.	.	.	1	92.6
repeat	2	.	1	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	1	1	1	3	1	1	.	1	.	3	.	240	1	.	.	1	1	.	1	.	93.8
seven	.	2	.	2	1	1	.	.	.	2	.	.	239	.	1	1	3	1	3	.	93.4
six	7	4	.	.	.	.	1	.	.	.	1	.	.	240	2	.	1	.	.	.	93.8
start	2	.	.	2	.	.	2	.	.	.	.	.	.	2	215	30	.	.	3	.	84.0
stop	3	.	.	.	.	.	12	.	.	.	.	2	.	.	18	221	.	.	.	.	86.3
three	5	1	1	2	1	.	.	.	.	.	.	.	4	2	.	.	224	16	.	.	87.5
two	8	.	.	.	.	3	5	.	2	.	.	.	9	2	.	.	1	222	1	3	86.7
yes	4	.	1	.	.	.	.	.	.	.	.	.	.	2	3	.	.	.	246	.	96.1
zero	5	3	1	.	.	13	3	.	11	.	.	1	.	.	.	.	3	13	.	203	79.3

Table 33.--Confusion matrix for one pass training

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	254	1	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	99.2
enter	2	251	1	.	.	.	.	.	.	.	.	.	.	.	.	.	1	1	.	.	98.0
erase	1	.	246	1	.	.	.	2	.	.	1	.	.	.	.	.	4	.	1	.	96.1
five	.	.	.	247	2	1	.	.	.	.	.	.	.	.	4	2	.	.	.	.	96.5
four	1	.	.	.	253	.	.	.	.	.	.	.	.	.	1	1	.	.	.	.	98.8
go	.	.	.	.	.	205	9	.	35	.	.	.	.	.	.	1	1	4	.	1	80.1
help	4	2	.	.	.	1	244	.	.	.	1	.	.	2	.	2	.	.	.	.	95.3
nine	1	1	.	.	.	.	.	243	.	10	.	.	1	.	.	.	.	.	.	.	94.9
no	.	.	.	.	.	34	2	.	215	1	.	1	1	.	.	1	.	1	.	.	84.0
one	.	.	.	.	.	.	.	2	.	251	.	.	2	.	.	.	.	.	.	1	98.0
repeat	4	.	1	.	.	.	.	.	.	.	251	.	.	.	.	.	.	.	.	.	98.0
rubout	.	.	.	.	.	1	.	1	.	.	1	248	.	.	.	4	.	.	.	1	96.9
seven	3	2	.	1	1	1	.	1	.	.	.	.	243	1	1	1	.	1	.	.	94.9
six	1	2	.	.	.	.	1	.	.	.	.	.	.	251	1	.	.	.	.	.	98.0
start	.	.	.	1	.	.	3	.	.	.	.	.	.	1	230	21	.	.	.	.	89.8
stop	2	2	.	.	.	.	8	.	.	.	.	.	.	.	12	232	.	.	.	.	90.6
three	5	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	236	11	3	.	92.2
two	4	.	.	.	.	1	2	.	.	.	.	.	1	.	.	.	1	247	.	.	96.5
yes	1	.	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	254	.	99.2
zero	3	2	.	.	1	10	2	.	8	.	1	2	.	.	.	.	1	13	.	213	83.2

Table 34.--Confusion matrix for recursive training

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	253	1	.	.	.	.	1	.	.	.	.	.	.	1	.	.	.	.	.	.	98.8
enter	2	252	2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	98.4
erase	1	.	246	1	.	.	.	2	.	.	1	.	.	.	.	.	4	.	1	.	96.1
five	.	.	.	246	.	1	.	.	.	.	.	.	.	.	8	1	.	.	.	.	96.1
four	1	.	.	.	255	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	99.6
go	.	1	.	.	.	217	6	.	25	.	.	.	.	.	.	.	.	3	.	4	84.8
help	4	2	.	.	.	.	243	.	1	.	1	.	.	2	.	2	.	1	.	.	94.9
nine	1	1	.	1	.	.	.	245	1	6	.	.	.	.	.	.	1	.	.	.	95.7
no	.	.	.	.	.	37	1	.	213	1	.	.	.	.	1	.	.	1	.	2	83.2
one	.	.	.	1	.	.	.	2	.	250	.	.	2	.	.	.	.	.	.	1	97.7
repeat	3	.	.	.	.	.	.	.	.	.	253	.	.	.	.	.	.	.	.	.	98.8
rubout	.	.	.	2	.	.	.	.	.	.	1	250	.	.	.	1	1	.	1	.	97.7
seven	4	3	.	1	.	.	.	.	.	.	.	.	243	1	1	1	1	1	.	.	94.9
six	1	2	.	.	.	.	1	.	.	.	1	.	.	250	1	.	.	.	.	.	97.7
start	.	.	.	1	.	.	3	.	.	.	.	.	.	1	232	19	.	.	.	.	90.6
stop	2	2	.	.	.	.	10	.	.	.	.	.	.	.	16	226	.	.	.	.	88.3
three	7	.	3	.	.	.	.	.	.	.	.	.	2	1	.	.	233	10	.	.	91.0
two	5	.	.	.	1	2	1	.	.	.	.	.	1	.	.	.	.	245	.	1	95.7
yes	2	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	.	.	253	.	98.8
zero	3	2	2	.	.	13	2	.	10	.	2	1	1	.	.	.	1	13	.	206	80.5

Table 35.--Confusion matrix for one sided vq

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e			o	r
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	252	1	.	.	.	.	1	.	.	.	.	.	.	1	.	.	.	1	.	.	98.4
enter	1	254	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	99.2
erase	1	.	246	1	.	.	.	3	.	.	1	.	.	.	.	.	3	.	1	.	96.1
five	.	.	.	247	.	1	.	.	.	.	.	.	.	.	4	4	.	.	.	.	96.5
four	1	.	.	.	253	.	.	.	.	.	.	.	.	.	2	.	.	.	.	.	98.8
go	.	.	.	.	.	220	10	.	19	.	.	.	.	.	.	1	.	4	.	2	85.9
help	3	2	.	.	.	.	248	.	.	.	1	.	.	2	.	.	.	.	.	.	96.9
nine	1	1	2	1	.	.	.	242	.	8	.	.	.	.	.	.	1	.	.	.	94.5
no	.	.	.	.	.	41	4	.	204	3	.	.	.	.	2	.	.	1	.	1	79.7
one	.	.	.	.	.	.	.	1	.	254	.	.	.	.	.	.	.	.	.	1	99.2
repeat	4	.	.	.	.	.	.	.	.	.	252	.	.	.	.	.	.	.	.	.	98.4
rubout	.	.	.	1	.	.	.	.	.	.	1	247	.	.	.	7	.	.	.	.	96.5
seven	1	2	.	1	1	.	2	.	.	.	.	.	244	1	2	.	1	1	.	.	95.3
six	1	2	.	.	.	.	2	.	.	.	1	.	.	250	.	.	.	.	.	.	97.7
start	.	.	.	.	.	.	3	.	.	.	.	.	.	1	232	20	.	.	.	.	90.6
stop	2	2	.	.	.	.	8	.	.	.	.	.	.	.	3	241	.	.	.	.	94.1
three	5	.	1	.	.	.	.	.	.	.	.	.	.	1	.	.	240	7	2	.	93.8
two	7	.	.	.	.	2	.	.	.	.	.	.	1	.	.	.	.	246	.	.	96.1
yes	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	255	.	99.6
zero	4	1	.	.	1	10	6	.	14	.	1	1	.	.	.	.	1	13	.	204	79.7

Table 36.--Confusion matrix for normalized distortion

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e		e	o	e		r	p	e			o	r	e
	t	r	e							a	u	n		t		e					(%)
eight	251	1	.	.	.	.	1	.	.	.	.	.	.	2	.	.	.	1	.	.	98.0
enter	1	254	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	1	.	.	99.2
erase	1	.	247	1	.	.	.	3	.	.	1	.	.	.	.	.	2	.	1	.	96.5
five	1	.	.	251	.	.	.	.	.	.	.	.	.	.	2	2	.	.	.	.	98.0
four	1	.	.	.	253	.	.	.	.	.	.	.	.	.	2	.	.	.	.	.	98.8
go	.	.	.	.	.	228	4	.	16	.	.	.	.	.	.	1	.	4	.	3	89.1
help	5	2	.	.	.	1	247	.	.	.	.	.	.	1	.	.	.	.	.	.	96.5
nine	1	1	2	1	.	.	.	244	.	7	.	.	.	.	.	.	.	.	.	.	95.3
no	.	.	.	.	.	43	2	1	203	1	.	.	1	.	1	.	.	1	.	3	79.3
one	.	.	.	1	.	.	.	2	.	252	.	.	.	.	.	.	.	.	.	1	98.4
repeat	4	.	.	.	.	.	.	.	.	.	252	.	.	.	.	.	.	.	.	.	98.4
rubout	.	.	.	1	.	.	.	.	.	.	1	246	.	.	.	8	.	.	.	.	96.1
seven	.	2	.	1	2	.	1	.	.	.	.	.	245	1	1	.	1	2	.	.	95.7
six	.	2	.	.	.	.	3	.	.	.	1	.	.	250	.	.	.	.	.	.	97.7
start	.	.	.	2	.	.	2	.	.	.	.	.	.	1	241	10	.	.	.	.	94.1
stop	2	2	.	.	.	.	8	.	.	.	.	.	.	.	6	238	.	.	.	.	93.0
three	1	.	2	.	.	.	.	.	.	.	.	.	.	1	.	.	244	6	2	.	95.3
two	4	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	251	.	.	98.0
yes	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	256	.	100.0
zero	4	.	.	.	.	7	2	.	12	.	2	1	.	.	.	.	1	12	.	215	84.0

Table 37.--Confusion matrix for cross speaker training

	e	e	e	f	f	g	h	n	n	o	r	r	s	s	s	s	t	t	y	z	s
	i	n	r	i	o	o	e	i	o	n	e	u	e	i	t	t	h	w	e	e	c
	g	t	a	v	u		l	n		e	p	b	v	x	a	o	r	o	s	r	o
	h	e	s	e	r		p	e			e	o	e		r	p	e		o	r	e
	t	r	e								a	u	n		t		e				e
											t	t									(%)
eight	253	1	.	.	.	.	1	.	.	.	.	.	.	1	.	.	.	.	.	.	98.8
enter	1	254	1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	99.2
erase	1	.	248	1	.	.	.	1	.	.	1	.	.	.	.	.	3	.	1	.	96.9
five	1	.	.	249	.	.	1	.	.	.	.	.	.	.	4	1	.	.	.	.	97.3
four	1	.	.	.	253	.	.	.	.	.	.	.	.	.	1	.	.	1	.	.	98.8
go	.	1	.	.	.	225	3	.	20	.	.	.	.	.	.	1	.	3	.	3	87.9
help	4	1	.	.	.	.	248	.	.	.	1	.	.	2	.	.	.	.	.	.	96.9
nine	1	1	1	.	.	.	.	248	.	3	.	.	.	.	.	.	2	.	.	.	96.9
no	.	.	.	.	.	46	2	.	196	2	.	1	1	.	1	.	.	2	.	5	76.6
one	.	.	.	.	.	.	.	2	.	251	.	.	1	.	.	.	.	.	1	1	98.0
repeat	4	.	.	.	.	.	.	.	.	.	252	.	.	.	.	.	.	.	.	.	98.4
rubout	.	.	.	2	.	.	.	.	.	.	.	244	1	.	.	8	.	.	.	1	95.3
seven	1	2	.	.	1	.	2	.	.	.	.	.	246	1	2	.	1	.	.	.	96.1
six	.	1	.	.	.	.	3	.	.	.	1	.	.	250	1	.	.	.	.	.	97.7
start	.	.	.	1	.	.	1	.	.	.	.	.	.	1	237	16	.	.	.	.	92.6
stop	2	2	.	.	.	.	9	.	.	.	.	.	.	.	3	240	.	.	.	.	93.8
three	2	.	1	.	.	.	.	.	.	.	.	.	.	1	.	.	244	7	1	.	95.3
two	3	.	.	.	.	1	.	.	.	.	.	.	.	.	.	.	.	252	.	.	98.4
yes	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	256	.	100.0
zero	4	1	.	.	.	13	1	.	9	1	1	2	.	.	.	.	.	9	1	214	83.6

BIBLIOGRAPHY

- Acoustical Society of America (1988). "Information for contributors to the Journal of the Acoustical Society of America," J. Acoust. Soc. Am. 83, introductory pages.
- Ainsworth, W. A., and Millar, J. B. (1971). "Methodology of experiments on the perception of synthesized vowels," Language and Speech 14, 201-212.
- American Chemical Society (1974). Bibliographic Guide for Editors and Authors (Am. Chem. Soc., Washington, D. C.).
- Applebaum, T. H., Hanson, B. A., and Wakita, H. (1987). "Weighted cepstral distance measures in vector quantization based speech recognizers," Proc. IEEE ICASSP 87, 1155-1158.
- Bahl, L. R., Das, S. K., de Souza, P. V., Jelinek, F., Katz, S., Mercer, R. L., Picheny, M. A. (1984). "Some experiments with large vocabulary isolated word sentence recognition," Proc. IEEE ICASSP 84, 26.5.1-26.5.2.
- Bell, C. G., Fujisaki, H., Heinz, J. M., Stevens, K. N., and House, A. S. (1961). "Reduction of speech spectra by analysis-by-synthesis techniques," J. Acoust. Soc. Am. 33, 1725-1736.
- Bergh, A. F., Soong, F. K., and Rabiner, L. R. (1985). "Incorporation of temporal structure into a vector quantization based preprocessor for speaker independent, isolated word recognition," AT&T Technical Journal 64, 1047-1063.
- Bonneau, H., and Gauvain, J. L. (1987). "Vector quantization for speaker adaptation," Proc. IEEE ICASSP 87, 1434-1437.

- Bourland, H., Wellekens, C. J., and Ney, H. (1984). "Connected digit recognition using vector quantization," Proc. IEEE ICASSP 84, 26.10.1-26.10.4.
- Bridle, J. S., Brown, M. D., and Chamberlain, R. M. (1983). "Continuous connected word recognition using whole word templates," The Radio and Electronics Engineer **53**, 167-175.
- Burton, D. K.. (1987). "Text dependent speaker verification using vector quantization source coding," IEEE Trans. Acoust. Speech Signal Process., **ASSP-35**, 133-143.
- Burton, D. K., Shore, J. E., and Buck, J. T. (1985). "Isolated-word speech recognition using multisection vector quantization codebooks," IEEE Trans. Acoust. Speech Signal Process., **ASSP-33**, 837-849.
- Bush, M. A., and Kopec, G. E. (1987). "Network based connected digit recognition," IEEE Trans. Acoust. Speech Signal Process. **ASSP-35**, 1401-1413.
- Buzo, A., Gray, A. H., Gray, R. M., and Markel, J. D. (1980). "Speech coding based upon vector quantization," IEEE. Trans. Acoust. Speech Signal Process., **ASSP-28**, 562-574.
- Chamberlin, H. (1985). Musical Applications of Microprocessors (Hayden Book Co., Berkeley, CA).
- Chomsky, N., and Halle, M. (1968). Sound Pattern of English (Harper and Row, New York).
- Compernelle, D. V. (1987). "Increased noise immunity in large vocabulary speech recognition with the aid of spectral subtraction," Proc. IEEE ICASSP 87, 1143-1146.

- Delattre, P., Liberman, A. M., Cooper, F. S., and Gerstman, L. J. (1952). "An experimental study of the acoustic determinants of vowel color," *Word* **8**, 195-210.
- Di Benedetto, M. G., and Lanaro, A. (1984). "How to avoid vowel normalization in identification of vowels in continuous speech," *Proc. IEEE ICASSP 84*, 42.4.1-42.4.4.
- Digital Sound Corporation. (1982). DSC 200 User Manual.
- Doddington, G. R., and Schalk, T. B. (1981). "Speech recognition: Turning theory to practice," *IEEE Spectrum* **18**(9), 26-32.
- Dodge, C., and Jerse, T. (1985). Computer Music (Schirmer Books, New York).
- Egan, J. P. (1948). "Articulation testing methods," *Laryngoscope* **58**, 955-991.
- Equitz, W. (1987). "Fast algorithms for vector quantization picture coding," *Proc. IEEE ICASSP 87*, 725-728; expanded paper submitted to *IEEE Trans. Acoust. Speech Signal Process.* (abstract published *ASSP-35*, 1361).
- Fant, G. (1970). Acoustic Theory of Speech Production (Mouton, The Hague, The Netherlands).
- Fant, G. (1973). Speech Sounds and Features (MIT Press, Cambridge, MA).
- Furui, S. (1987). "A vq based preprocessor using cepstral dynamic features for large vocabulary word recognition," *Proc. IEEE ICASSP 87*, 1127-1130.
- Gersho, A., and Cuperman, V. (1983). "Vector quantization: A pattern matching technique for speech coding," *IEEE Commun. Magazine*, December, 15-21.

- Glinski, S. C. (1985). "On the use of vector quantization for connected digit recognition," AT&T Technical Journal **64**, 1033-1045.
- Golibersuch, R. J. (1983). "Automatic prediction of linear frequency warp for speech recognition," Proc. IEEE ICASSP 83, 769-772.
- Gray, R. M. (1984). "Vector quantization," IEEE ASSP Magazine **1**(2), 4-29.
- Heinz, J. M., and Stevens, K. N. (1961). "On the properties of voiceless fricative consonants," J. Acoust. Soc. Am. **33**, 589-596.
- Hess, W. (1983). Pitch Determination of Speech Signals (Springer-Verlag, New York).
- Hieronymus, J. L., and Majurski, W. J. (1985). "A reference speech recognition algorithm for benchmarking and speech data base analysis," Proc. IEEE ICASSP 85, 1573-1576.
- IBM Speech Recognition Group. (1985). "A real-time, isolated word, speech recognition system for dictation transcription," Proc. IEEE ICASSP 85, 858-861.
- IBM Speech Recognition Group. (1987). "Experiments with the Tangora 20,000 word speech recognizer," Proc. IEEE ICASSP 87, 701-704.
- Kaegi, W., and Tempelaars, S. (1978). "VOSIM -- a new sound synthesis system," J. Audio Eng. Society **26**, 418-425.
- Kernighan, B. W. and Ritchie, D. M. (1978). The C Programming Language (Prentice-Hall, Englewood Cliffs, NJ).
- Klatt, D. H. (1974). "Word verification in a speech understanding system," in Speech Recognition, edited by D. R. Reddy (Academic Press, New York), pp. 321-341.

- Klatt, D. H. (1977). "Review of the ARPA Speech Understanding Project," J. Acoust. Soc. Am. **62**, 1345-1366; reprinted (1980) in Trends in Speech Recognition, edited by W. A. Lea (Prentice-Hall, Englewood Cliffs, NJ), pp. 249-271.
- Kopec, G. (1986). "Formant tracking using hidden Markov models and vector quantization," IEEE Trans. Acoust. Speech Signal Process. **ASSP-34**, 709-729.
- Kopec, G. E., and Bush, M. A. (1985). "Network based isolated digit recognition using vector quantization," IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**, 850-867.
- Landell, B. P., Naylor, J. A., and Wohlford, R. E. (1984). "Effect of vector quantization on a continuous speech recognition system," Proc. IEEE ICASSP 84, 26.11.1-26.11.4.
- Lee, D. L., and Lochovsky, F. H. (1983). "Voice response systems," ACM Computing Surveys **15**, 351-374.
- Levinson, S. E. (1985). "Structural methods in automatic speech recognition," Proc. IEEE **73**, 1625-1650.
- Liberman, A. M. (1970). "The grammars of speech and language," Cognitive psychology **1**, 301-323.
- Licklider, J. C. R., Bindra, D., and Pollack, I. (1948). "The intelligibility of rectangular speech waves," Am. J. Psychol. **61**, 1-20.
- Linde, Y., Buzo, A., and Gray, R. M. (1980). "An algorithm for vector quantizer design," IEEE Trans. Commun. **COM-28**, 84-95.
- Lloyd, S. P. (1957). "Least squares quantization in PCM," Bell Lab. Tech. Note; reprinted in IEEE Trans. Inf. Theory **IT-28** (1982), 129-137.

- Loeb, E. P., and Lyon, R. F. (1987). "Experiments in isolated digit recognition with a cochlear model," Proc. IEEE ICASSP 87, 1131-1134.
- Mergel, D., and Ney, H. (1985). "Phonetically guided clustering for isolated word recognition," Proc. IEEE ICASSP 85, 854-857.
- Miller, R. L. (1953). "Auditory tests with synthetic vowels," J. Acoust. Soc. Am. **25**, 114-121.
- Moore, R. K., Russell, M. J., and Tomlinson, M. J. (1983). "The discriminative network; a mechanism for focusing recognition in whole word pattern matching," Proc. IEEE ICASSP 83, 1041-1044.
- Myers, C. S., and Rabiner, L. R. (1981). "A level building dynamic time warping algorithm for connected word recognition," IEEE Trans. Acoust. Speech Signal Process. **ASSP-29**, 284-297.
- National Bureau of Standards, "The Texas Instruments isolated word data base - 46 word version - PCM/VCR format."
- Ney, H. (1984). "The use of a one stage dynamic programming algorithm for connected word recognition," IEEE Trans. Acoust. Speech Signal Process. **ASSP-32**, 194-196.
- Ney, H., Mergel, D., and Marcus, S. M. (1986). "On the automatic training of phonetic units for word recognition," IEEE Trans. Acoust. Speech Signal Process. **ASSP-34**, 209-213.
- O'Shaughnessy, D. (1987). Speech Communication: Human and Machine (Addison-Wesley, Reading, MA).
- Olson, H. F., and Belar, H. (1956). "Phonetic typewriter," J. Acoust. Soc. Am. **28**, 1072-1081.
- Paliwal, K. K., and Rao, P. V. S. (1982). "Synthesis based recognition of continuous speech," J. Acoust. Soc. Am. **71**, 1016-1024.

- Pallet, D. S. (1985). "Performance assessment of automatic speech recognizers," J. Res. Nat. Bur. Stand. **90**, 371-387; reprinted (1986) in Electronic Speech Recognition, edited by G. Bristow (McGraw-Hill, New York), pp. 277-306.
- Pallet, D. S. (1986). "A benchmark for speaker dependent recognition using the Texas Instruments 20 word and alpha-set speech database," presented at DARPA Speech Recognition Workshop, Feb. 86; available from National Bureau of Standards.
- Pan, K. C., Soong, F. K., and Rabiner, L. R. (1985). "A vector quantization based preprocessor for speaker independent isolated word recognition," IEEE Trans. Acoust. Speech Signal Process. **ASSP-33**, 546-560.
- Parsons, T. (1987). Voice and Speech Processing (McGraw-Hill, New York).
- Peterson, G. E., and Barney, H. L. (1952). "Control methods used in a study of the vowels," J. Acoust. Soc. Am. **24**, 175-184.
- Pieraccini, R., and Billi, R. (1983). "Experimental comparison among data compression techniques in isolated word recognition," Proc. IEEE ICASSP 83, 1025-1028.
- Rabiner, L. R. (1984). "On the application of energy contours to the recognition of connected word sequences," AT&T Bell Laboratories Technical Journal **63**, 1981-1995.
- Rabiner, L. R., and Soong, F. K. (1985). "Single frame vowel recognition using vector quantization with several distance measures," AT&T Technical Journal **64**, 2319-2330.
- Rabiner, L. R., Levinson, S. E., and Sondhi, M. M. (1983). "On the application of vector quantization and hidden Markov models to speaker independent, isolated word recognition," Bell Syst. Tech. J. **62**, 1075-1105.

- Rabiner, L. R., Sondhi, M. M., and Levinson, S. E. (1983). "Note on the Properties of a Vector Quantizer for LPC Coefficients," Bell Syst. Tech. J. **62**, 2603-2616.
- Rabiner, L. R., Sondhi, M. M., and Levinson, S. E. (1984). "A vector quantizer incorporating both LPC shape and energy," Proc. IEEE ICASSP 84, 17.1.1-17.1.4.
- Rabiner, L. R., Sondhi, M. M., and Levinson, S. E. (1984). "On the use of hidden Markov models for speaker independent recognition of isolated words from a medium size vocabulary," AT&T Bell Laboratories Technical Journal **63**, 627-641.
- Reininger, H., and Wolf, D. (1985). "Speech and speaker independent codebook design in VQ coding schemes," Proc. IEEE ICASSP 85, 1700-1702.
- Roe, D. B. (1987). "Speech recognition with a noise adapting codebook," Proc. IEEE ICASSP 87, 1139-1142.
- Roucos, S., and Wilgus, A. M. (1985). "The waveform segment vocoder: A new approach for very low rate speech coding," Proc. IEEE ICASSP 85, 236-239.
- Scott Instruments Corporation. (1986). Coretechs VET3 Speech Terminal User Manual.
- Scott, B. L. and Hardesty, L. H. (1983). "Method and apparatus for speech recognition," United States Patent no. 4,383,135.
- Scott, B. L., Goodman, R. G., Newell, J. M., and Smith, L. A. (1987). "Method and apparatus for generating a signal transformation and the use thereof in signal processing," United States patent application, pending.
- Shannon, C. E. (1948). "A mathematical model of communication," Bell Syst. Tech. J. **27**, 379-423, 623-656.

- Shore, J. E., and Burton, D. K. (1983). "Discrete utterance speech recognition without time alignment," *IEEE Trans. Inf. Theory*, **IT-29**, 473-491.
- Silvertsen, E. (1961). "Segment inventories for speech synthesis," *Language and Speech* **4**, 27-89.
- Soong, F. K., Rosenberg, A. E., and Juang, B. H. (1987). "A vector quantization approach to speaker recognition," *AT&T Technical Journal* **66**, 14-26.
- Sugamura, N., Shikano, K., and Furui, S. (1983). "Isolated word recognition using phoneme like templates," *Proc. IEEE ICASSP 83*, 723-726.
- Thosar, R. B., and Rao, P. V. S. (1976). "An approach towards a synthesis based speech recognition system," *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-24**, 194-196.
- Voiers, W. D. (1979). "Intelligibility testing at Dynastat: The diagnostic rhyme test," Dynastat Inc., Austin, TX.
- Witten, I. H. (1982). Principles of Computer Speech (Academic Press, New York).