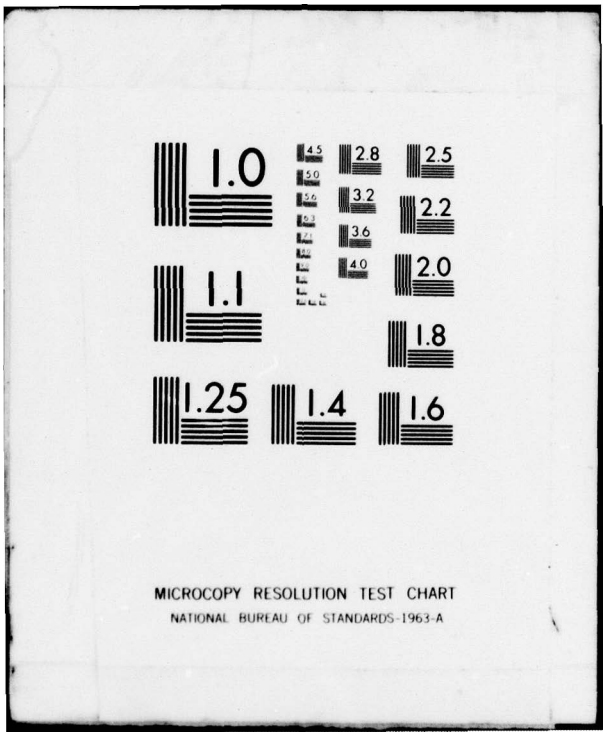
AD-A074 206 TEXAS INSTRUMENTS INC DALLAS F/G 17/2
LIMITED VOCABULARY CONTINUOUS WORD RECOGNITION.(U)
AUG 79 R L DAVIS, B G SECREST, C J CATO F30602-77-C-0139
UNCLASSIFIED RADC-TR-79-204 NL

1 OF 2
AD
A074206


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

RADC-TR-79-204
Final Technical Report
August 1979

LEVEL II (12)

LIMITED VOCABULARY CONTINUOUS WORD RECOGNITION

Texas Instruments, Inc.

Robert L. Davis
Bruce G. Secrest
Craig J. Cato

DDC
RECEIVED
SEP 25 1979
E

AD A074206

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

DDC FILE COPY

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

79 09 24 001

This report has been reviewed by the RADC Information Office (OI) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-79-204 has been reviewed and is approved for publication.

APPROVED: Richard S. Vonusa

RICHARD S. VONUSA
Project Engineer

APPROVED:

ROSS H. ROGERS, Colonel, USAF
Chief, Intelligence & Reconnaissance Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRAA), Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RADC-TR-79-204	2. GOVT ACCESSION NO.
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE *(and Subtitle)*: LIMITED VOCABULARY CONTINUOUS WORD RECOGNITION	5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report, August 1977 — January 1979
	6. PERFORMING ORG. REPORT NUMBER: N/A
7. AUTHOR(s): Robert L. Davis, Bruce G. Secrest, Craig J. Cato	8. CONTRACT OR GRANT NUMBER(s): F30602-77-C-0139
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Texas Instruments Inc., 13500 North Central Expressway, Dallas TX 75265	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 40270803
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (IRAA), Griffiss AFB NY 13441	12. REPORT DATE: August 1979
	13. NUMBER OF PAGES: 121
14. MONITORING AGENCY NAME & ADDRESS *(if different from Controlling Office)*: Same	15. SECURITY CLASS. *(of this report)*: UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Richard S. Vonusa (IRAA)

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Speech Processing
Digit Recognition
Word Recognition
Pattern Recognition
Clustering

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The objective of this contract was to develop the capability of recognizing a limited vocabulary (20 words) from continuous speech, independent of a limited number of speakers. The research during this study contract was concentrated in three major areas:

(1) High-performance, speaker-independent, connected-digit recognition for syntactically unconstrained digit sequences. (Cont'd)

DD FORM 1 JAN 73 1473

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

347 650

(2) Clustering algorithms for use in the development of sets of reference patterns for speaker-independent word recognition.

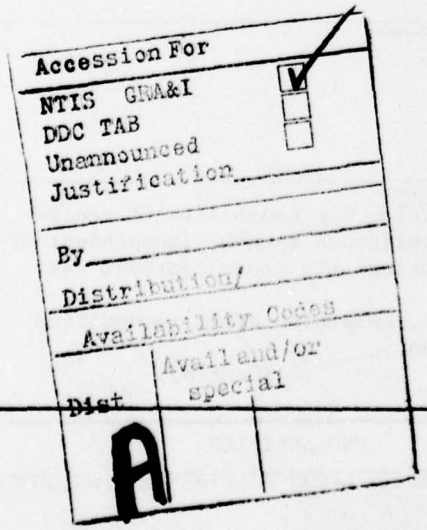
(3) Automatic enrollment for speaker-dependent, connected-word recognition for syntactically unconstrained word sequences of 20 words.

The program culminated in the installation of the speaker-independent, connected-digit recognition program on the Base and Installation Security System's advanced development model speaker verification system at RADC.

The speaker-independent, connected-digit recognition portion of this study resulted in a significantly faster algorithm with a 50-percent decrease in error rate over the course of this study-from 9.5 percent to 4.7 percent on an evaluation data set of 10 six-digit sequences from 106 speakers (64 males, 42 females).

The development of the clustering algorithm resulted in a two-stage, four-path algorithm with the mechanisms for detecting outlying data points in the design data and with subsequent analysis routines for comparing the results from the various paths and testing the validity of resulting clusters on the basis of comparisons with a priori information about the design data set.

The research into development of an automatic enrollment technique for speaker-dependent word recognition resulted in a method that yielded very good results for isolated word recognition but less acceptable results when used in continuous speech from the same speaker. The better results achieved with comparable hand enrollments point to the need for continued development of automated enrollment, and the desirability of an interim solution of a semiautomated enrollment procedure allowing the operator the option of modifying reference-point locations and recognition-pattern format definitions defined by an automated front end. Independent of the enrollment method, however, the benefit of reference file updating as a means of accommodating contextual variability, as well as intersession variability, was abundantly clear.



EVALUATION

The objective of this program was to develop techniques and algorithms to extend highly reliable speaker-dependent isolated word recognition to speaker-dependent continuous word recognition and study the methodology for speaker-independent continuous speech recognition.

A hardware/software implementation of a real-time continuous speech recognition system was fabricated by Texas Instruments (TI). This system was extensively tested and modified to incorporate the results of the tests and a continual upgrade of the system took place over the life of the contract. TI based their real-time speech recognition system on techniques they developed for automatic speaker verification and the Total Voice Verification program which used a restricted connected digit capability.

The speaker-independent, connected digit performance resulted in 95.3 percent recognition accuracy on a data set consisting of 10 six-digit sequences from 106 speakers (64 males and 42 females).

The capability that TI has developed under subject program has been installed at RADC for further test and evaluation. The RADC tests shall attempt to establish the effectiveness of the current state-of-the-art connected speech system as to its applicability to operational military requirements.

Richard Vonusa

RICHARD S. VONUSA
Project Engineer

TABLE OF CONTENTS

Section	*Title*	*Page*
I	INTRODUCTION	1
II	CONTINUOUS SPEECH RECOGNITION	3
	A. Speech Representation	3
	B. Time Normalization	10
	C. Reference-Point Location	15
	D. Word Hypothesizing and Testing	17
	E. An Efficient Tree Searching Algorithm	19
	F. Overall Word-Recognition Algorithm	24
III	SPEAKER-DEPENDENT WORD RECOGNITION	27
	A. Introduction	27
	B. Enrollment	27
	C. Updating	29
	D. Application to Continuous Speech	29
IV	REFERENCE PATTERN GENERATION FOR SPEAKER-INDEPENDENT WORD RECOGNITION	33
	A. Review of Clustering in Speaker-Independent Word Recognition	33
	B. Detailed Clustering Algorithm	37
	C. Criteria for Measuring Partition Goodness	40
	D. Description of Cluster Analysis Documentation	42
	1. Trees	42
	2. Parameter Comparisons	42
	3. Consistency Tests	42
	E. Observations on Clustering Results	50
	F. Testing Cluster Validity With *a priori* Information	53
V	GENERAL-PURPOSE SPEECH I/O CAPABILITY	75
	A. System Description	75
	B. Filter Simulation in AP120B	76
	C. Voicing Decision From the AP-120B Autocorrelation Pitch Tracker	76
VI	EXPERIMENTAL RESULTS	77
	A. Speaker-Independent Digit Recognition	77
	1. Data Sets	77
	2. Digit Recognition Results for Six-Digit Sequences	77
	3. Digit Recognition Results for Three-Digit Sequences	83
	4. Digit Recognition Results for Isolated Digits	88
	5. Effect of Spectral Normalization Technique on Digit Recognition Performance	89
	B. Limited Vocabulary Word-Recognition Experiment	93
VII	CONCLUSIONS AND RECOMMENDATIONS	97
	REFERENCES	

APPENDIXES

A. Speech Processing
B. Scatter Matrices
C. tr S_B Alternate Form Derivation
D. Postiterative Optimization Statistics for Recognition Patterns

LIST OF ILLUSTRATIONS

Figure	*Title*	*Page*
1	Hierarchical Organization of Feature Abstraction Network H	3
2	*Quantized Spectral Speech Representation*	10
3	Demonstration of Need for Time Alignment Between Spectra for Two "Sevens"	11
4	Relationship of Piecewise Linear Time Normalization to Nonlinear Time Normalization	12
5	Example of Recognition Pattern Formation	14
6	Example of Scanning Pattern Formation	16
7	Example of Valley Finding	18
8	Example of Directed Graph	19
9	Flow Chart for Efficient Tree-Searching Algorithm	23
10	Word Recognition Algorithm Flow Chart	25
11	Automatic Enrollment for "Two"	30
12	Automatic Enrollment for "Six"	32
13	Bell Laboratories Clustering Procedures	35
14	Block Diagram of Clustering Procedures Developed for Clustering Scanning and Recognition Patterns for Speaker-Independent Digit Recognition	38
15	Tree for Final 24 Stages of MINAVE Agglomerative Clustering of All (166) Scanning Patterns for First Reference Point of Digit "Zero"	44
16	Parameter Comparisons for Pre- and Postiterative Optimization of MINAVE Partitions Using All Points	45
17	Parameter Comparisons for Pre- and Postiterative Optimization of MINAVE Partitions With Outliers Discarded	46
18	Parameter Comparisons for Pre- and Postiterative Optimization of MINMAX Partitions Using All Points	47
19	Parameter Comparisons for Pre- and Postiterative Optimization of MINMAX Partitions With Outliers Discarded	48
20	Parameter Comparisons for Preiterative Optimization of MINAVE and MINMAX Partitions Using All Points	49
21	Parameter Comparisons for Postiterative Optimization of MINAVE and MINMAX Partitions Using All Points	50
22	Class Assignments and Contingency Tables for Postiterative Optimization of 10 Classes Using All Points for Reference Point 1 of Digit 0	51
23	Contingency Table Measures for Pre- and Postiterative Optimization Using All Points for Reference Point 1 of Digit 0	52
24	Histograms of Normalized Entropy as Measure of Dispersion in Class Sizes for MINAVE Agglomerative Clustering of Scanning Patterns	54

25 Histograms of Normalized Entropy as Measure of Dispersion in Class Sizes for MINAVE Agglomerative Clustering of Recognition Patterns 55
26 Histograms of Normalized Entropy as Measure of Dispersion in Class Size for Preiterative Optimization of Recognition Patterns 56
27 Histograms of Normalized Entropy as Measure of Dispersion in Class Sizes for Preiterative Optimization of Recognition Patterns 56
28 $\sigma_{post}/\sigma_{pre}$ Versus $\alpha_{post}/\alpha_{pre}$ for MINAVE Agglomerative Clustering of Scanning Patterns 59
29 $\sigma_{post}/\sigma_{pre}$ Versus $\alpha_{post}/\alpha_{pre}$ for MINMAX Agglomerative Clustering of Scanning Patterns 60
30 σ Versus α for Preiterative Optimization of MINAVE Agglomerative Clusters of Scanning Patterns 61
31 σ Versus α for Preiterative Optimization of MINMAX Agglomerative Clusters of Scanning Patterns 62
32 σ Versus α for Postiterative Optimization of MINAVE Agglomerative Clusters of Scanning Patterns 63
33 Comparison of Mutual Information for Clustered Scanning Patterns Using Four Clustering Algorithms 70
34 Comparison of Mutual Information for Clustering Recognition Patterns Using Four Clustering Algorithms 73
35 Speech Channel Block Diagram 75
36 Scanning Pattern Format 84
37 Spectra for Digit "Two" for Speaker J.S. 90

LIST OF TABLES

Table *Title* *Page*

1 Characteristics of 16-Channel Filter Bank 9
2 Recognition Pattern Format Definitions for Digits 15
3 Sample of Information Contained in Table of Hypothesized Digits 20
4 Sample of Table of Backpointers for Best Word Sequences in Sorted Table of Hypothesized Words 21
5 Values of Seven Clustering Criteria for Postiterative Optimization of MINMAX Agglomerative Clusters for Recognition Pattern for Digit "Six" 41
6 Criterion Values for Tree for Final 24 Stages of MINAVE Agglomerative Clustering for First Reference Point of "Zero" 43
7 Contingency Table of First-Kind Results for Preiterative Optimization 57
8 Contingency Table of First-Kind Results of Postiterative Optimization 58
9 Mutual Information and Residues for Preiterative Optimization of MINAVE Agglomerative Clusters of Scanning Patterns 66
10 Mutual Information and Residues for Postiterative Optimization of MINAVE Agglomerative Clusters of Scanning Patterns 67
11 Mutual Information and Residues for Preiterative Optimization of MINMAX Agglomerative Clusters of Scanning Patterns 68
12 Mutual Information and Residues for Postiterative Optimization of MINMAX Agglomerative Clusters of Scanning Patterns 69
13 Mutual Information and Residues for Preiterative Optimization of MINAVE Agglomerative Clusters of Recognition Patterns 71

14 Mutual Information and Residues for Postiterative Optimization of MINAVE Agglomerative Clusters of Recognition Patterns . . . 71
15 Mutual Information and Residues for Preiterative Optimization of MINMAX Agglomerative Clusters of Recognition Patterns . . . 72
16 Mutual Information and Residues for Postiterative Optimization of MINMAX Agglomerative Clusters of Recognition Patterns . . . 72
17 Ten Sets of Ten Six-Digit Sequences Used in Testing . . . 78
18 Martin-Herscher Digit Texts . . . 79
19 Synopsis of Evaluation Results . . . 80
20 Digit Recognition Results for Selected Evaluation Runs . . . 82
21 T-Function Quantization Thresholds . . . 85
22 Decrease in Average Recognition Pattern Error by Including T-Function in Scanning Patterns . . . 85
23 Confusion Matrix for Digit Recognition for 6-Digit Sequences for Run No. 62 . . . 85
24 Recognition Error (TE) Normalizing Constants . . . 86
25 Confusion Matrix for Digit Recognition for 6-Digit Sequences for Run No. 73 . . . 86
26 Digit-Recognition Performance of Males and Females . . . 86
27 Histogram of Digit-Recognition Performance . . . 87
28 Confusion Matrix for Digit Recognition for 3-Digit Sequences Constrained in Length . . . 88
29 Confusion Matrix for Digit Recognition of Isolated Digits . . . 89
30 Total Normalized Error (NE) for Digits From Sequence 852-734 for Speaker J.S. . . . 91
31 Valley Point Errors, Sequence Errors (SQ), and Recognition Errors (TE) for Digits From Sequence 852-734 for Speaker J.S. . . . 91
32 σ_{min} Variation Performance Test . . . 92
33 Confusion Matrix for Automatic Enrollment . . . 94
34 Confusion Matrix for Automatic Enrollment . . . 95
35 Percent Correct Speaker-Dependent Recognition Results for Continuous Utterances From a 20-Word Vocabulary Enrolled on Isolated Words . . . 96

SECTION I
INTRODUCTION

This final report covers the research done on a limited-vocabulary continuous word recognition study undertaken by Texas Instruments. This effort was divided into two primary areas of investigation: extension of speaker-dependent isolated word recognition to speaker-dependent continuous word recognition, and the study of speaker-independent continuous speech recognition.

Speaker-dependent isolated word recognition is currently being used for applications such as map data entry. Extension to speaker-dependent continuous word recognition is a more natural one for the time normalization techniques used at Texas Instruments (described in Section II) than the type that depends on locating endpoints of words, which may not even exist [e.g., when phonemes are shared (/s/ in six-seven) between words in continuous speech]. Speaker-dependent word recognition uses speaker-dependent reference patterns obtained in a single enrollment session. A method of automatic enrollment and supervised updating to accommodate intersession variations and context dependencies were investigated during this study.

For many years, the approach to the problem of speaker-independent recognition of continuous speech has been a heuristically directed search for the correct features and weightings for the hierarchical classification of a set of symbol strings, mapping ultimately into an English-language transcription. The emphasis has been on getting out of the acoustic and into the phonemic domain as quickly as possible because of the huge memory requirements for storing acoustic data for large vocabularies. Since the heuristics were often based on the researcher's judgment, derived from often insufficient data, the consequent mislabelings had to be corrected with progressively more complex classification algorithms. Design and testing using small data bases, along with the use of phonemic representations of speech have resulted not only from memory limitations but also from the lack of techniques in speech for dealing with very large amounts of data. Within the last few years, however, work on such techniques has begun to appear. During the Total Voice Speaker Verification study,[1] performed by Texas Instruments under RADC sponsorship, a clustering algorithm was developed and used to produce a set of speaker-independent reference patterns for use in speaker-independent, connected-digit recognition. The current study then concentrated on two tasks for speaker-independent, continuous-speech recognition. One task was to determine the performance that could be achieved on speaker-independent, connected-digit recognition using the previously developed reference patterns by making improvements to the word recognition algorithm. The other task was to investigate improvements that could be made to the clustering algorithm for the purpose of finding better partitions of the design data set.

Section II of this report reviews the speech technology used at Texas Instruments and covers an improved directed graph searching algorithm developed during this contract. Section III covers an automated enrollment method for speaker-dependent, connected-word recognition and the role of reference-pattern updating. Section IV describes the application of clustering to speaker-independent reference-pattern generation and covers the algorithm extensions developed

[1] R.L. Davis, B.M. Hydrick, and G.R. Doddington, "Total Voice Speaker Verification," Rome Air Development Center Technical Report, RADC-TR-78-260, January 1978.

during this contract. Section V covers several general-purpose speech-processing capabilities that center on the use of direct speech input/output (I/O) to a fast array processor. The experimental results for both the extensive testing performed for speaker-independent, connected-digit recognition and the more limited testing done for speaker-dependent, continuous-speech recognition are covered in Section VI. Conclusions and recommendations are made in Section VII.

SECTION II

CONTINUOUS SPEECH RECOGNITION

During the relatively short history of continuous speech recognition work, the classification schemes have used a feature abstraction process from the speech waveform followed by a hierarchical classifier. The level of abstraction varied from features of the waveform itself to symbol representations (phonemes) requiring highly sophisticated classification techniques in order to compensate for segmentation and labeling errors. The classification complexity generally was proportional to the level of abstraction. Martin[2] shows a tree (Figure 1) of feature abstraction levels.

The usual argument for using a symbol is that it offers a more compact representation of words and, hence, growth in the memory requirement is not so dramatic with increase in vocabulary size. However, as Reddy[3] points out, good signal-to-symbol transformation techniques currently do not exist, causing size increases in the lexicons and the algorithms, not only to account for context, dialect, and idolect variations, but also to account for mislabeled acoustic events.

Therefore, reference-pattern matching in the signal domain has the advantage of not having to accommodate feature abstraction errors. Three crucial problems are involved, however: selection of the speech representation, time normalization of the speech signal for matching with reference patterns, and selection of the reference patterns themselves. The first two of these topics are discussed in the remainder of this section and the reference-pattern selection is the subject of the more extended discussions in Sections III and IV.

A. SPEECH REPRESENTATION

The specific speech representation used in this study was the output of a 16-channel digital filter bank preceded by a first-order differencing network (for preemphasis). Each of the bandpass filters is a two-section, cascaded, second-order Bessel filter followed by a rectifier and a lowpass filter sampled every 10 milliseconds (ms). Center frequencies and bandwidths for the 16 filters are given in Table 1.

An important consideration is the choice of wide-bandwidth filters that locate spectral peaks but that avoid resolving the voice fundamental and its harmonics. Note that the center frequencies for filters 14 through 16 lie in the part of the spectrum primarily occupied by energy only during fricatives. The exception is the third formant for the vowel /i/ for males and for all the front vowels for females. Since no precise resolution of the frequency location is possible with the wide-bandwidth filters, the only interest is the presence or absence of a third formant (in which case other formants would also exist in lower filters) or the presence or absence of energy anywhere in the frequency band of the top three filters without lower frequency energies. In order to compact the filter bank representation, the top three filters were added into one value, without averaging because of the depressed amplitudes, yielding a 14-element vector to represent the speech spectrum:

[2]T.B. Martin, "Acoustic Recognition of a Limited Vocabulary in Continuous Speech," Ph.D. Dissertation, University of Pennsylvania, 1970.

[3]D.R. Reddy, "Speech Recognition by Machine: A Review," *Proceedings of the IEEE,* 64:501–531, April 1976.

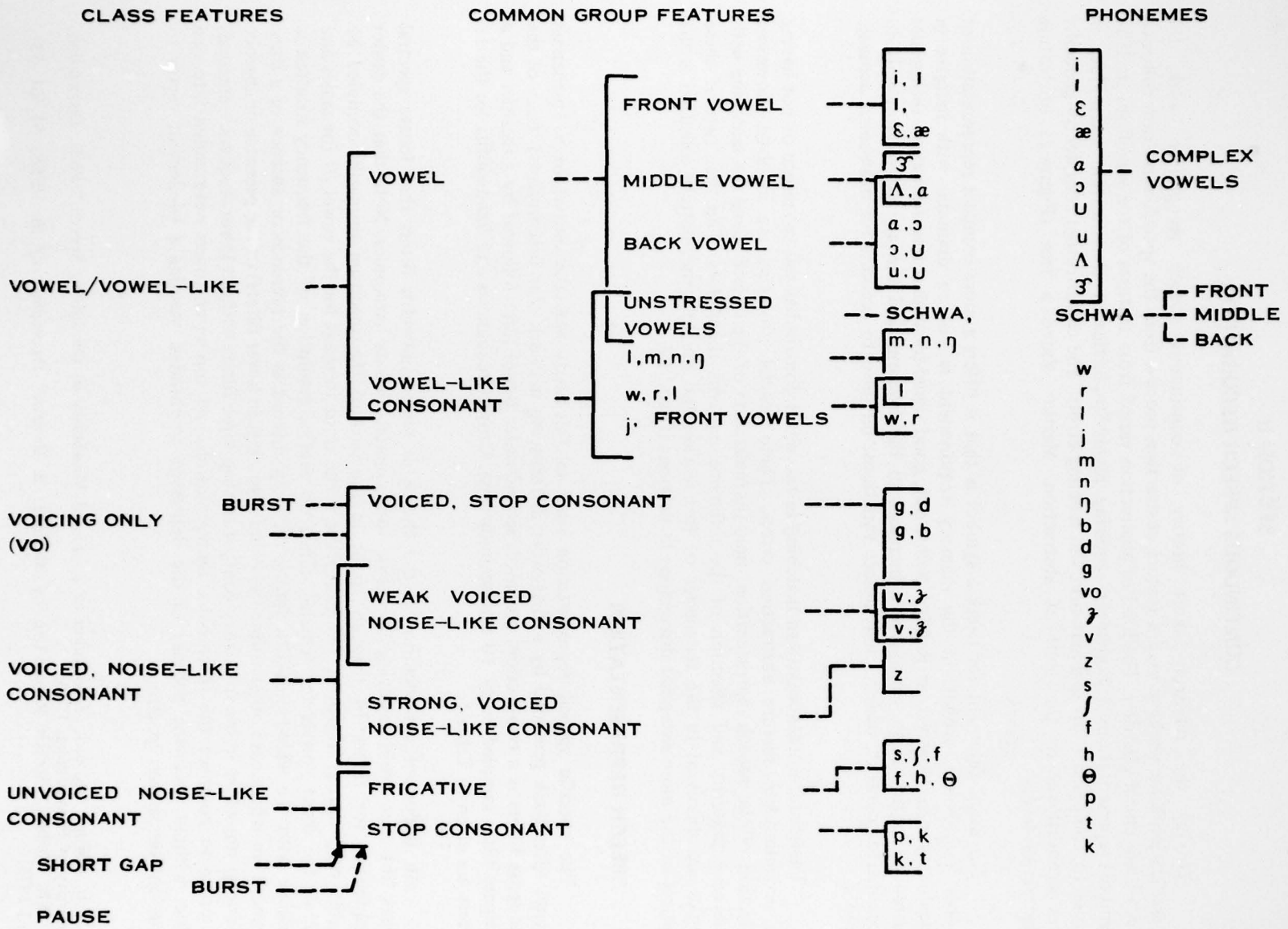


Figure 1. Hierarchical Organization of Feature Abstraction Network (From Martin[2])

$$\vec{A}_j = \begin{Bmatrix} a_{1j} \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{14j} \end{Bmatrix} = \begin{Bmatrix} a_1(t_j) \\ a_2(t_j) \\ \cdot \\ \cdot \\ \cdot \\ a_{14}(t_j) \end{Bmatrix}$$

This 14-element vector is regressed (Appendix A) using a sine and a cosine basis function to eliminate gross aspects of the spectrum and to flatten the spectrum. The two regression coefficients, c_1 and c_2, along with a measure of the energy in filters 2 through 13 (vowel energy), are concatenated to the 14 regressed filter outputs. All elements except the energy are then normalized and quantized to one of eight equiprobable values, resulting in a speech representation such as that shown in Figure 2 for the word "seven." The form shown in Figure 2 is used throughout the remainder of this report. The values of the normalized, quantized a_{ij} and the two regression coefficients are indicated by the density of the printed symbols according to the following:

Value:	0	1	2	3	4	5	6	7
Symbol:	blank	,	"	+	=	O	B	$

At this point the energy is not quantized; however, it is always used relative to other energies and the relative value is then quantized. Further detail of the speech representation can be found in the Total Voice Speaker Verification study final report.[1]

TABLE 1. CHARACTERISTICS OF 16-CHANNEL FILTER BANK

Filter	Center Frequency (Hz)	Bandwidth (Hz, at −6 dB)
1	280	250
2	395	280
3	525	310
4	630	340
5	750	360
6	900	360
7	1080	360
8	1265	365
9	1480	365
10	1725	365
11	1985	365
12	2285	360
13	2640	365
14	3150	625
15	3720	635
16	4235	615

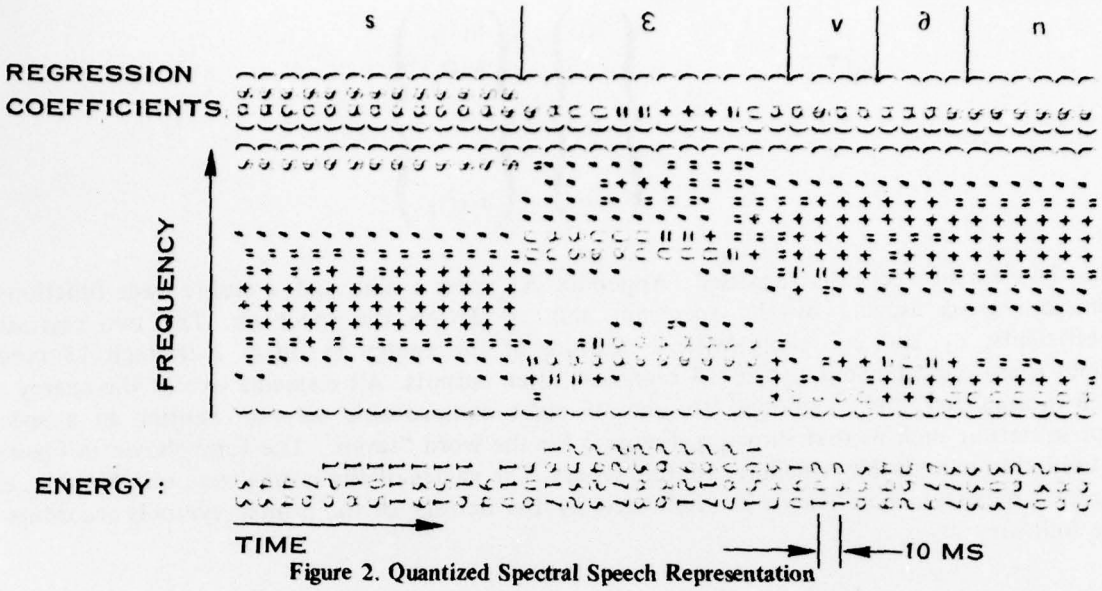


Figure 2. Quantized Spectral Speech Representation

B. TIME NORMALIZATION

One of the basic problems in speech processing is time alignment of the speech waveform with respect to a reference. For example, in the two spectrograms in Figure 3 for the word "seven," the time differences between corresponding Δs (which denote phonemic boundaries) are obvious.

Early work used linear time normalization of two patterns between endpoints of words, and although this method improved recognition performance, it suffered from an inability to deal with the nonlinear fluctuations between endpoints and to locate endpoints in continuous speech reliably.

Two distinct approaches developed during the late 1960s and early 1970s. One approach (most of the ARPA sponsored work: Reddy[3]) was based on translating a string of input features into a sequence of phonemic labels, a procedure dependent on accurate segmentation between phonemes. Segmentation and labeling errors were then repaired by more sophisticated subsequent processing using syntax, semantics, etc.

The other method approached the problem by a nonlinear warping of the time axis of a feature waveform of the input speech to obtain maximum coincidence with a reference waveform. This approach was used by both Doddington[4] and Sakoe and Chiba;[5] however, the latter's approach could be more easily represented in a form amenable to the use of dynamic programming, useful in easing the computation burden. This dynamic programming approach has

[4]G.R. Doddington, "A Method of Speaker Verification," Ph.D. Dissertation (Thesis), University of Wisconsin, 1970.

[5]H. Sakoe and S. Chiba, "A Dynamic Programming Approach to Continuous Speech Recognition," *Proceedings of the 7th International Congress on Acoustics*, August 1971.

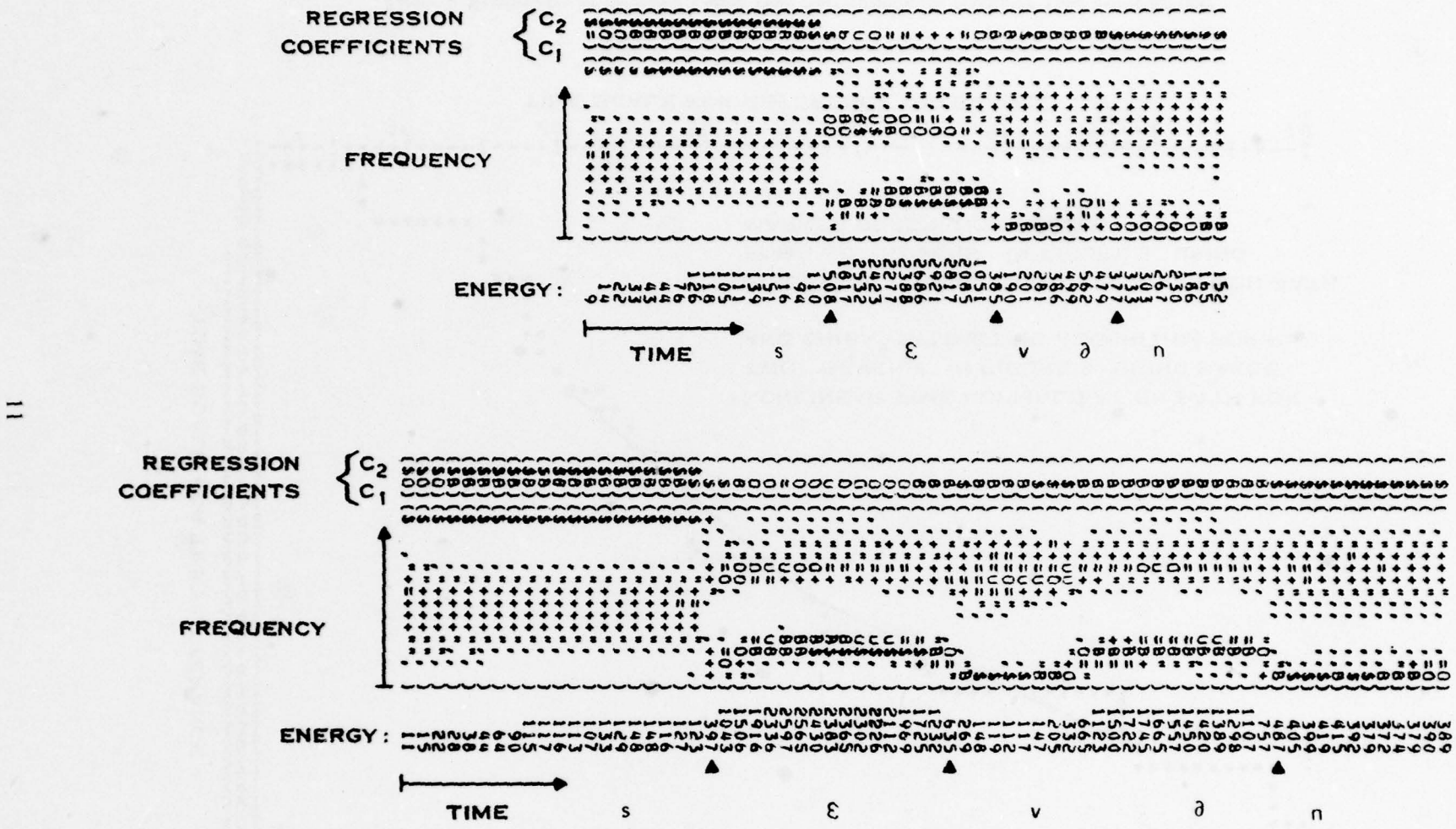


Figure 3. Demonstration of Need for Time Alignment Between Spectra for Two "Sevens"

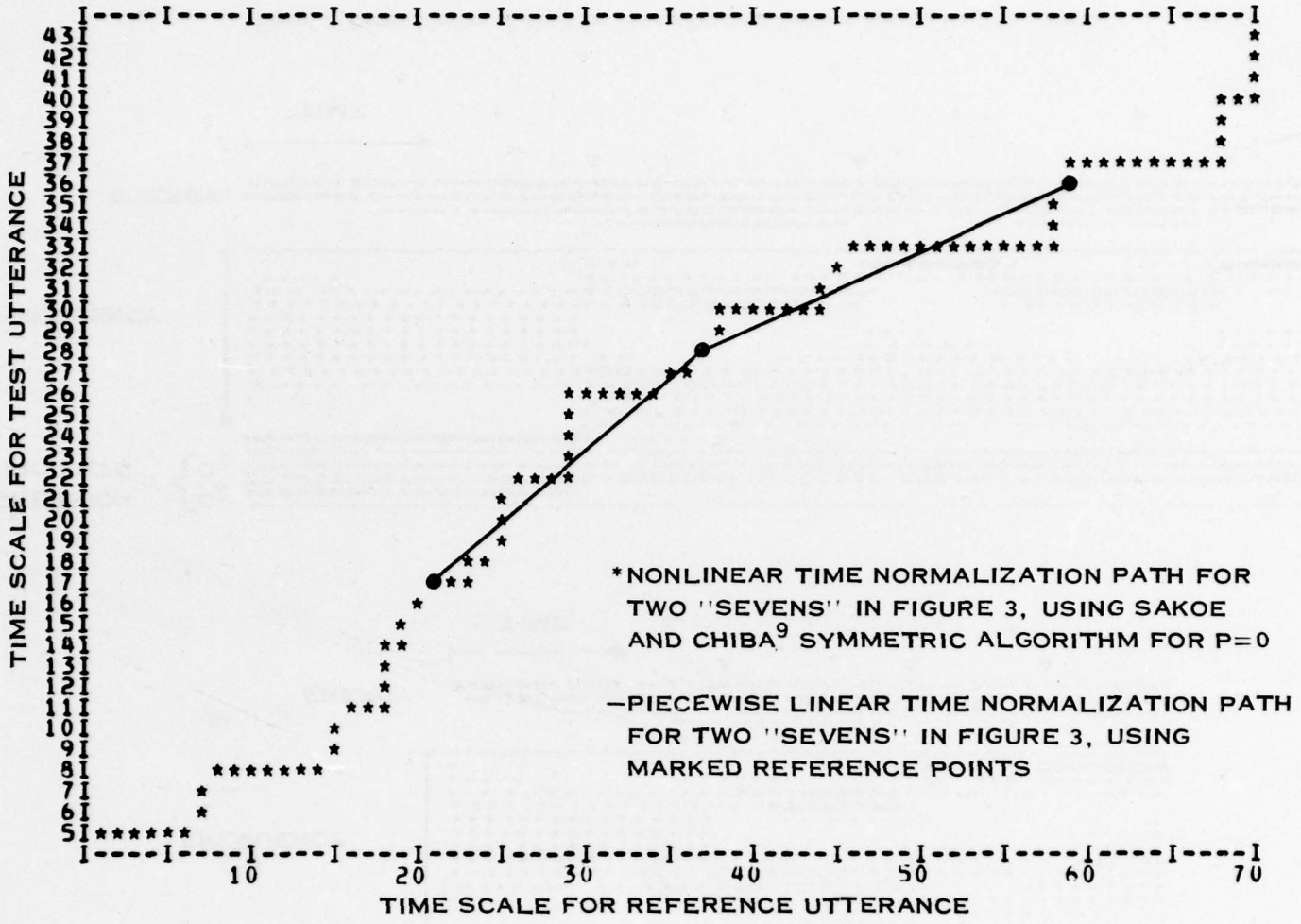


Figure 4. Relationship of Piecewise Linear Time Normalization to Nonlinear Time Normalization

been used by Velichko and Zagoruiko,[6] Itakura,[7] White and Neely,[8] and Sakoe and Chiba[9] in isolated word recognition. Extension to continuous-word recognition has been done by Lowerre[10] on the HARPY speech recognition system (also described briefly by White[11]), by Porter,[12] and by Nippon Electric Company in their DP-100 Connected Speech Recognition System.

The technique used at Texas Instruments is an amalgamation of these two methods and was first used by Doddington[13] in 1973. In this method, potential acoustic boundaries (reference points) are first located in the input waveform. Reference points are combined into optimal sequences for words in the vocabulary using a dynamic programming routine that uses a measure of how reliably the reference points were located and that accounts for the deviations from expected time differences between reference points.

After sequences of potential reference points have been identified, the input waveforms are interpolated linearly between reference points to form a time normalized representation of the utterance. The relationship to the Sakoe/Chiba approach can be seen in Figure 4. Essentially, only those points along the path of the time warp that represent acoustic boundaries are found (the o's in Figure 4), and the linear interpolation is then performed between these reference points. A piecewise linear, time-normalized, acoustic representation of the word (a "recognition pattern") is thus formed. A sample of the spectral data portion of a recognition pattern being extracted from input speech spectra is shown in Figure 5.

As an example of the choice of reference-point locations, the reference points (Δs) for the digits are shown in Table 2 for the phonetic transcriptions of the general American dialect pronunciations for the digits as found in Kenyon and Knott.[14] These locations were chosen at points that would exhibit large spectral changes. The actual rules used in extracting recognition patterns for the 10 digits are also specified in table 2, where:

(1) Initial negative numbers indicate the columns for extrapolation before the first reference point

(2) Intermediate numbers in parentheses indicate the number of columns for interpolation between reference points

(3) The remaining numbers indicate columns for extrapolation after the last reference point.

[6] V.M. Velichko and N.G. Zagoruiko, "Automatic Recognition of 200 Words," *International Journal Man-Machine Studies*, 2:223, June 1970.

[7] F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-23:67–72, February 1975.

[8] G.M. White and R.B. Neely, "Speech Recognition Experiments With Linear Prediction, Bandpass Filtering, and Dynamic Programming," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-24:183–188, April 1976.

[9] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-26:43–49, February 1978.

[10] B.T. Lowerre, "The HARPY Speech Recognition System," Ph.D. Dissertation (Thesis), Carnegie-Mellon University, 1976.

[11] G.M. White, "Continuous Speech Recognition: Dynamic Programming, Knowledge Nets and HARPY," Paper 28-2, *1978 WESCON Professional Program*, September 1978.

[12] J.E. Porter, "LISTEN: A System for Recognizing Connected Speech Over Small, Fixed Vocabularies in Real Time," Naval Training Equipment Center Technical Report, NAVTRAEQUIPCEN 77-C-0096-1, April 1978.

[13] G.R. Doddington, "Speaker Verification," Rome Air Development Center Technical Report, RADC-TR-74-179, April 1974.

[14] J.S. Kenyon and T.A. Knott, *A Pronouncing Dictionary of American English*, G. & C. Merriam Company (Springfield, Massachusetts, 1953).

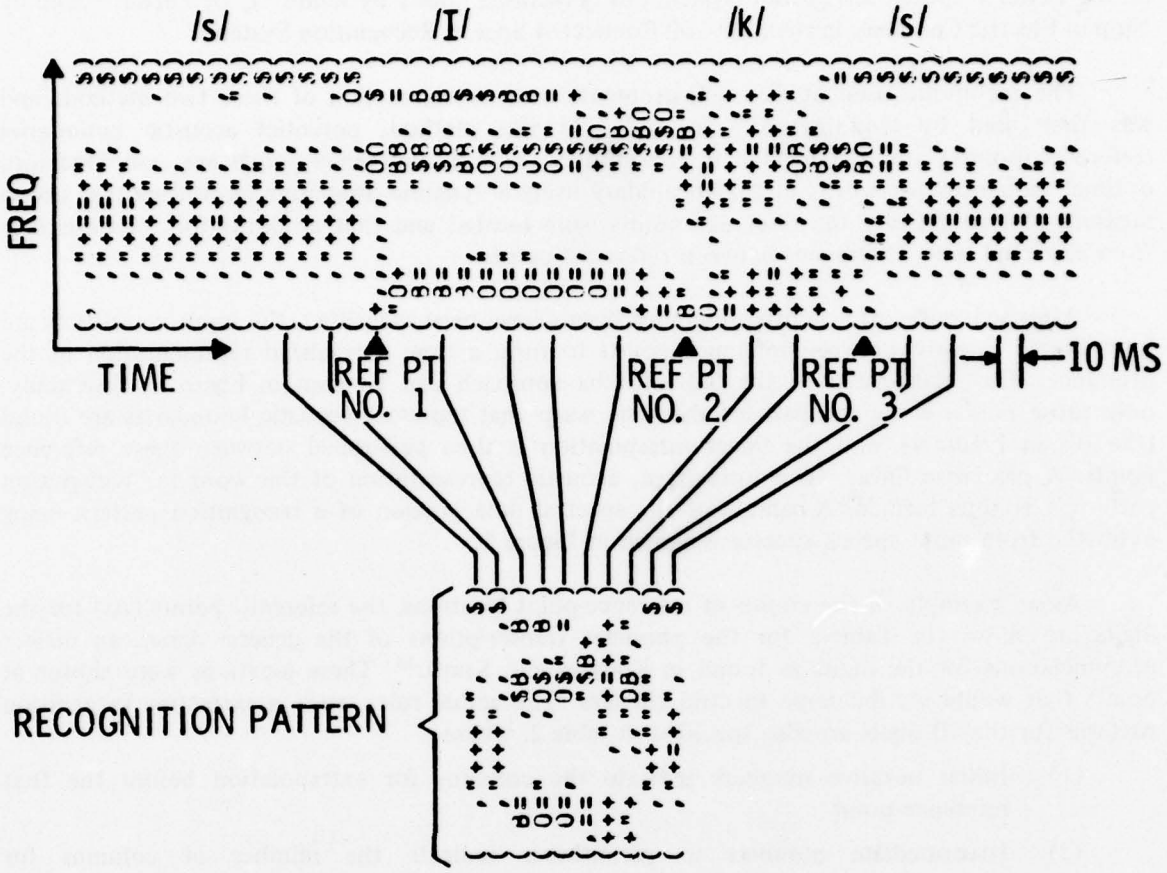


Figure 5. Example of Recognition Pattern Formation

At this point, the speech representation is still in the acoustic domain, differing from those[10,15] who transform their time-warped segments into phonemic labels with associated transition probabilities between labeled states. The advantage of remaining in the acoustic domain is that it avoids an intermediate classification that would introduce errors and obviates the need to find every phonetic boundary, which is helpful when such boundaries are difficult to find.

[15]F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proceedings of the IEEE*, 64:532–556, April 1976.

TABLE 2. RECOGNITION PATTERN FORMAT DEFINITIONS FOR THE DIGITS

z Δ I r Δ o Δ −4, −2 (4) (4)	f Δ a I Δ v −4, −2 (6) 2, 4
w Δ Λ Δ n −4, −2 (6) 2, 4	s Δ I Δ k Δ s −4, −2 (4) (2) 2, 4
Δ t Δ u Δ −4, −2 (2) (6)	s Δ ϵ Δ v ð Δ n −4 −2 (4) (4) 2, 4
θ Δ r i Δ −4, −2 (6)	Δ e Δ t (6) 2, 4
f Δ o Δ r −4, −2 (6) 2	n Δ a I Δ n −4, −2 (6) 2, 4

C. REFERENCE-POINT LOCATION

A presupposition of this piecewise, linear, time-normalization technique is extremely accurate reference-point location. One approach would be vocabulary-independent, locating changes in features such as voicing, energy, or spectrum between adjacent time samples. This is a reliable, precise method for use in speaker-dependent recognizers; however, sometimes expected acoustic segmentation points are missed in speaker-independent recognition.

A more robust approach is to use a vocabulary-dependent approach (similar to the "transeme" approach used at IBM[16]), matching a feature vector (called a "scanning pattern") extracted from the input speech waveform to reference scanning patterns, or templates. Figure 6 shows a scanning pattern being extracted from the spectral input. Matching is performed by computing a distance between the input and all reference patterns for every frame (10 ms in this study). Minima in this distance function are locations of potential acoustic boundaries (reference points).

More specifically, the scanning pattern formed at time t_j consists of: (1) the spectral data, regression coefficients, and energy for the five time samples from t_{j-2} through t_{j+2} and (2) the difference between the data for all adjacent pairs of time samples. The energy used in the scanning pattern is the energy for each of the five columns of data, normalized by the sum over all five columns and quantized to 4 bits. Figure 6 illustrates the formation of a scanning pattern from preprocessed (regressed, normalized, quantized) speech data. The only purpose of the difference data is to weight more heavily rapid changes of the feature vectors with respect to time. Since these data are derived from the standard data portion of the scanning pattern, subsequent illustrations of scanning patterns in this report will not show the difference data, even though it is, in fact, part of the actual pattern.

[16]N.R. Dixon and H.F. Silverman, "The 1976 Modular Acoustic Processor (MAP)," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-25:367–379, October 1977.

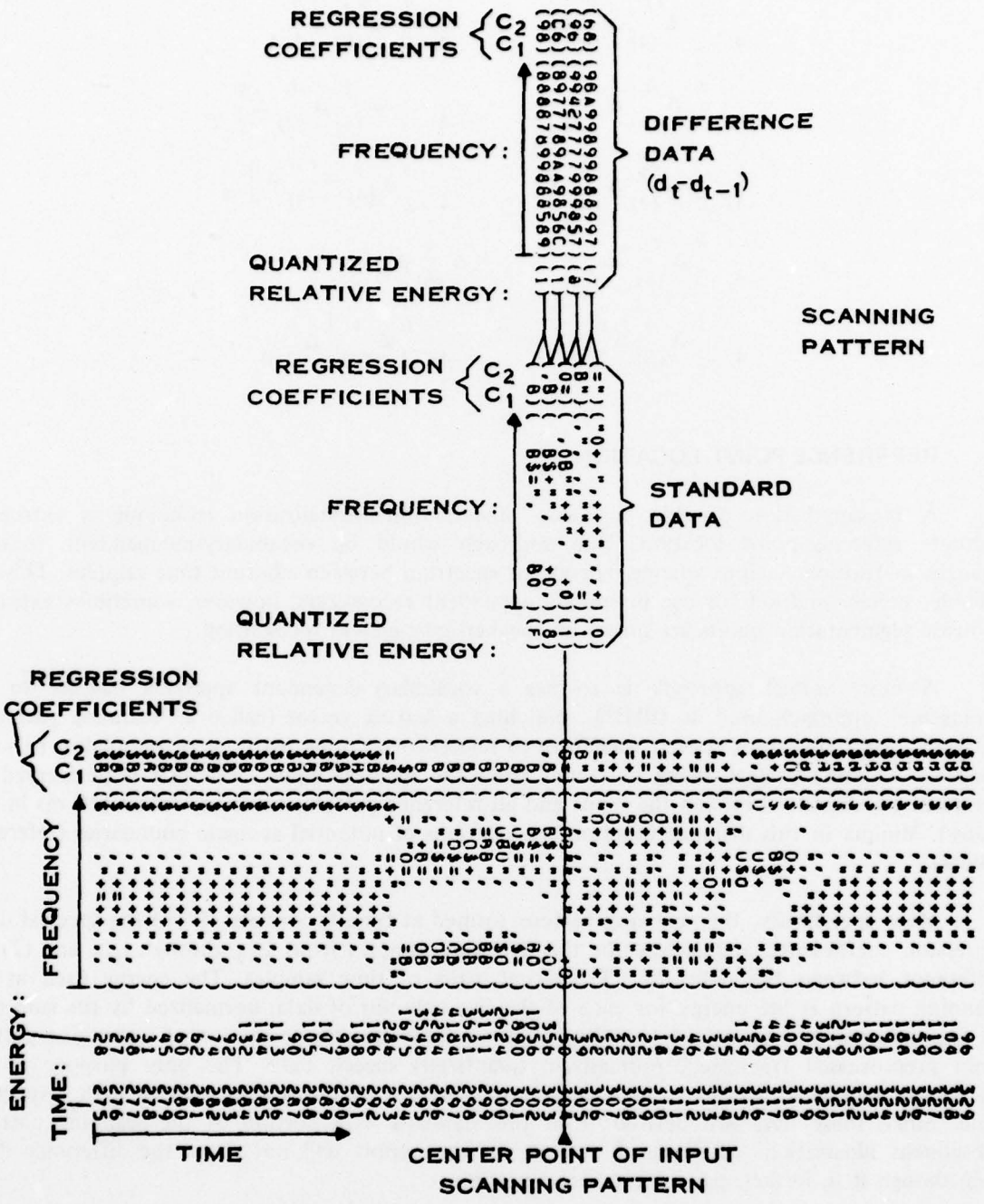


Figure 6. Example of Scanning Pattern Formation

In order to determine where reference points occur in the input speech, the input data are compared with reference data. This procedure (called scanning) is done by formatting scanning patterns from the input speech at each time sample t_j, comparing these with predetermined reference scanning patterns $\vec{r}_k$, and obtaining a measure of squared difference between the two, called the scanning error:

$$e_{kj} = ||\vec{x}_j - \vec{r}_k||^2 = \sum_{i=1}^{164} (x_{ij} - r_{ik})^2 \qquad (1)$$

The final error associated with each reference point is the minimum error of all comparisons with patterns representing that reference point.

Using the scanning errors as a function of time, an error function is thus generated for each type of reference scanning pattern using the minimum scanning error for each pattern type for each time sample. (Multiple reference scanning patterns may be allowed for each reference point of each word.) Each function is monitored for dips of sufficient magnitude to be considered as potential locations of the corresponding reference points in the input data. These dips are called valley points when the ratio of the scanning error following the dip to the scanning error at the dip itself is greater than or equal to a specified peak-to-valley ratio (PVR), typically 1.1 to 1.3, and the magnitude of the scanning error for the valley point is less than or equal to a threshold, typically 600 to 1,200. The occurrence of a peak (verified when the ratio of the scanning error following the peak to the scanning error at the peak is less than the reciprocal of the PVR) is required before another valley point can be found. The valley-finding procedure is shown in Figure 7.

D. WORD HYPOTHESIZING AND TESTING

Once these valleys in the scanning error (potential reference points) have been found, the next task is to fit them together to form word hypotheses. A sequence of time-ordered reference-point hypotheses for a word must exist, and the time distance between each pair of reference points must satisfy word-specific minimum/maximum restrictions. The error determined for each reference point pair is weighted by deviations from the expected distance between the two points and the scanning error at each hypothesized reference point. The weighted error for reference points i and j is:

$$E_{w_{i,j}} = \frac{(e_i + \text{offset})\,(e_j + \text{offset})}{1024} \left[1 + \beta \left(\frac{\hat{dt}_{i,j} - dt_{i,j}}{\hat{dt}^*_{i,j}} \right)^2 \right] \qquad (2)$$

where

$dt_{i,j} = t_j - t_i$ $\qquad$ $\beta = 2$

$\hat{dt}_{i,j}$ = expected $dt_{i,j}$ $\qquad$ $\hat{dt}_{min} = 4$

$\hat{dt}^*_{i,j} = \max(\hat{dt}_{i,j}\ \hat{dt}_{min})$ $\qquad$ offset = 100

e_i, e_j = scanning error for reference points i, j

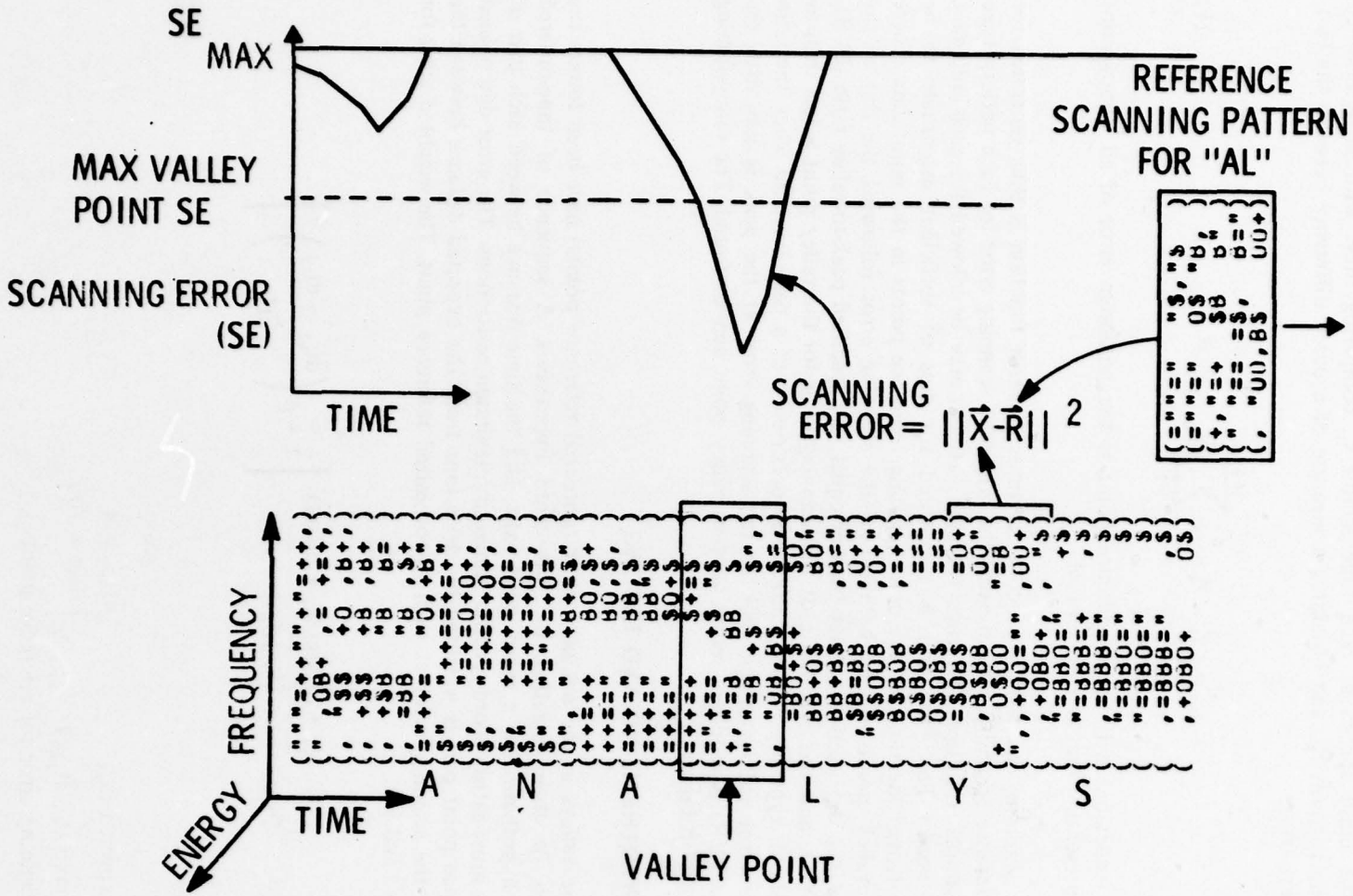


Figure 7. Example of Valley Finding

If the hypothesized word sequence error (SQ), which is the sum of the $E_{wi,j}$ for all reference point pairs in the word, is less than a predetermined word-specific threshold, then a word has been hypothesized.

To test this hypothesized word, the time-normalized recognition pattern anchored at the corresponding hypothesized reference points is compared to time-normalized reference recognition pattern(s) for the hypothesized word, using the squared Euclidian distance. This distance (or the minimum distance, in the case of multiple reference patterns) is the recognition error (TE), which is used along with the sequence error (SQ) in computing a total normalized error (NE) for the word "k" as given below:

$$NE_k = \frac{TE_k / \text{No. of columns in word k}}{\text{normalizing constant for word k}} + w_k \frac{SQ_k}{10\ (NPP)} \tag{3}$$

where NPP is the number of reference-point pairs used in computing SQ_k. If the NE for the hypotheiszed word is above a predefined threshold or if the average energy across the recognition pattern is less than a threshold, the word is discarded. Otherwise, it is placed into a table of hypothesized words, along with the SQ, TE, NE, average energy, reference-point times, and scanning errors for that word. Table 3 is an example table of hypothesized words for a sequence of six digits spoken continuously. Note the existence of the 35 superfluous words in this case. The optimal searching of this table (or directed graph) is described in the next subsection.

E. AN EFFICIENT TREE-SEARCHING ALGORITHM

Once the current speech segment has been completed, the sorted table of hypothesized words must be searched to find the sequence having the minimum error. Note that this corresponds to finding the best path through a directed graph, such as the simple one shown in Figure 8, taken from Porter.[12] The correspondence to the directed graph can be seen most readily from a specific example. Table 4 shows the table of hypothesized words (digits, in this case) sorted according to the time (in centiseconds) of the final reference points. Each entry in this table can point back in time to all previous digits with earlier final reference-point times (less than the initial reference-point time of the entry being considered). The allowable range of back-pointers can be limited by requiring that the time difference between the first reference point of each word and the last reference point of preceding words lie between a specified minimum and maximum.

Clearly, the exhaustive search time for traversing all possible paths increases rapidly.

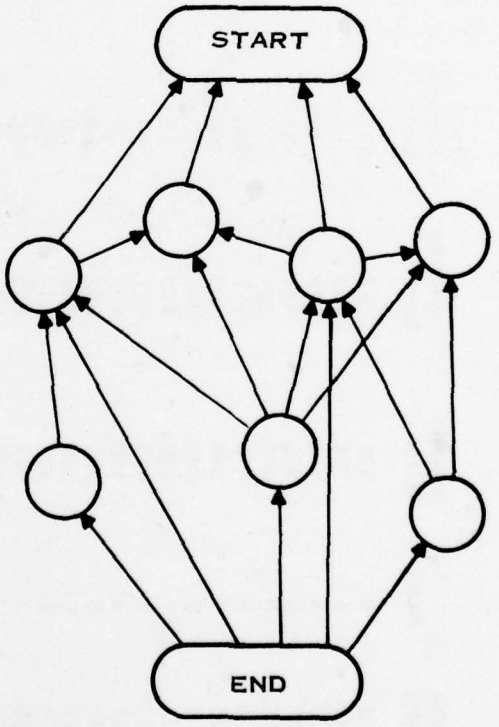


Figure 8. Example of Directed Graph

TABLE 3. SAMPLE INFORMATION CONTAINED IN TABLE OF HYPOTHESIZED DIGITS

Table Index	Word	Sequence Error	Recognition Error	Total Normalized Error	Normalized Error Threshold	Average Energy	Ref. Pt. 1 Time	Ref. Pt. 1 Error	Ref. Pt. 2 Time	Ref. Pt. 2 Error	Ref. Pt. 3 Time	Ref. Pt. 3 Error
41	9	522	1,055	112	114	1,283	415	295	454	448		
40	5	244	808	80	113	908	439	345	469	180		
39	4	162	315	59	116	1,173	439	256	464	131		
38	3	716	511	99	110	1,406	439	382	454	622		
37	0	495	598	77	128	1,148	439	501	455	296	469	161
36	9	576	951	106	114	646	394	272	435	493		
35	9	544	993	108	114	776	417	361	435	493		
34	9	446	998	103	114	875	415	295	436	477		
33	9	400	940	96	114	595	396	270	430	423		
32	9	452	931	99	114	916	415	295	430	423		
31	3	450	405	73	110	886	409	344	430	394		
30	9	274	907	88	114	787	394	272	422	277		
29	9	368	888	90	114	785	401	399	422	277		
28	9	438	932	98	114	736	401	427	423	319		
27	7	993	857	106	109	870	373	256	391	254	412	382
26	5	602	994	113	113	756	405	601	423	332		
25	3	346	337	58	110	1,168	409	344	423	265		
24	2	857	625	100	123	854	404	438	408	402	423	386
23	1	318	811	82	97	1,145	411	354	422	260		
22	0	631	734	95	128	840	374	405	394	187	413	519
21	8	374	361	64	107	510	387	463	400	219		
20	7	324	487	53	109	1,094	373	256	381	263	392	139
19	5	532	1,002	111	113	784	374	638	399	229		
18	3	394	408	70	110	1,000	374	427	398	215		
17	2	571	494	75	123	672	369	251	373	451	400	220
16	1	362	959	96	97	1,011	376	425	394	240		
15	0	542	482	66	128	1,116	374	405	386	230	392	231
14	9	578	823	96	114	2,054	343	246	351	399		
13	8	390	503	82	107	1,464	332	323	358	358		
12	6	494	531	83	110	952	331	263	355	324	362	112
11	3	360	381	65	110	1,406	331	329	359	318		
10	3	428	462	78	110	1,788	342	422	359	318		
9	3	486	395	72	110	1,586	331	329	355	377		
8	2	535	381	61	123	1,043	325	300	330	313	359	312
7	2	563	446	70	123	1,240	325	300	330	313	353	338
6	0	472	670	83	128	1,622	332	437	343	172	357	273
5	5	270	446	51	113	649	291	345	313	205		
4	9	384	888	91	114	1,120	250	396	268	289		
3	8	170	353	53	107	1,664	255	290	271	124		
2	5	432	916	99	113	1,067	250	392	271	348		
1	3	340	537	83	110	1,180	248	426	271	187		

TABLE 4. SAMPLE OF TABLE OF BACKPOINTERS FOR BEST WORD SEQUENCES IN SORTED TABLE OF HYPOTHESIZED WORD

Sorted Table Index	Time of First Ref. Pt.	Time of Last Ref. Pt.	Adjusted Normalized Error	Backpointer Lists for Sequences of Length 1	2	3	4	5	6
Best Sequence:				16	16	16	16	28	39
41	439	469	80	−1	16	16	16	16	28
40	439	469	69	−1	16	16	16	16	28
39	439	464	59	−1	16	16	16	16	28
38	415	454	112	−1	16	16	16	16	−1
37	439	454	99	−1	16	16	16	16	28
36	415	436	103	−1	16	16	16	16	−1
35	394	435	106	−1	5	5	11	−1	−1
34	417	435	108	−1	16	16	16	16	−1
33	396	430	96	−1	16	16	16	16	−1
32	415	430	99	−1	16	16	16	16	−1
31	409	430	73	−1	16	16	16	16	−1
30	401	423	98	−1	16	16	16	16	−1
29	405	423	113	−1	16	16	16	16	−1
28	409	423	58	−1	16	16	16	16	−1
27	404	423	90	−1	16	16	16	16	−1
26	394	422	88	−1	5	5	11	−1	−1
25	401	422	90	−1	16	16	16	16	−1
24	411	422	82	−1	16	16	16	16	−1
23	374	413	85	−1	5	5	11	−1	−1
22	373	412	95	−1	5	5	11	−1	−1
21	387	400	64	−1	5	5	11	−1	−1
20	369	400	67	−1	5	5	11	−1	−1
19	374	399	111	−1	5	5	11	−1	−1
18	374	398	70	−1	5	5	11	−1	−1
17	376	394	96	−1	5	5	11	−1	−1
16	373	392	47	−1	5	5	11	−1	−1
15	374	392	59	−1	5	5	11	−1	−1
14	331	362	74	−1	5	5	−1	−1	−1
13	331	359	64	−1	5	5	−1	−1	−1
12	342	359	78	−1	5	5	−1	−1	−1
11	325	359	54	−1	5	5	−1	−1	−1
10	332	358	82	−1	5	5	−1	−1	−1
9	332	357	74	−1	5	5	−1	−1	−1
8	331	355	72	−1	5	5	−1	−1	−1
7	325	353	63	−1	5	5	−1	−1	−1
6	343	351	96	−1	5	5	−1	−1	−1
5	291	313	51	−1	4	−1	−1	−1	−1
4	255	271	53	−1	−1	−1	−1	−1	−1
3	250	271	99	−1	−1	−1	−1	−1	−1
2	248	271	83	−1	−1	−1	−1	−1	−1
1	250	268	91	−1	−1	−1	−1	−1	−1

Syntactic constraints, such as those used in the total voice speaker verification study, can sometimes aid the efficiency of sequence finding, depending on whether the constraints apply locally to word pairs or more globally to the utterance as a whole. If syntactic constraints have been imposed in order to increase sequence recognition performance, such as in the total voice application, these constraints can be incorporated into the tree-searching algorithm to eliminate searching branches that are not syntactically correct.

An even greater potential saving may be obtained by saving optimal subsequences to avoid repetitious searching of the same path. However, this technique is not acceptable for sequences that are syntactically constrained since the saved optimal subsequence may not, in fact, satisfy the constraints when those constraints are applied to the entire sequence. In other words, the "correct" sequence (satisfying the syntactic constraints) may in fact not be the "best" (lowest error) sequence. Since the saving of optimal subsequences only finds one best sequence of a given length, this method is not appropriate to syntactically constrained word sequences.

For unconstrained sequences, however, this technique of saving optimal subsequences can shorten the exhaustive search time to be proportional only to the number of table entries preceding a table entry. As an example, Table 4 gives the resulting subsequence backpointers for the sorted table of hypothesized digits for the six-digit sequence in Table 3. These backpointer lists are constructed from the bottom up, according to the algorithm shown in Figure 9. Essentially, the backpointer for sorted table index i, sequence length k, points to the sorted table index j ($j < i$) that has the minimum error for a subsequence of length $(k - 1)$ for all j satisfying the relation:

$$\Delta t_{min} \leq (t_i^{initial} - t_j^{final}) \leq \Delta t_{max} \tag{4}$$

where Δt_{min} = 3 cs and Δt_{max} = 120 cs were used in this study.

As each new entry in the list of backpointers is constructed, it is compared to the best (lowest total error) subsequence of the same length. If the newer entry has a lower error, this error replaces the best error, and the pointer to the best sequence of that length is changed to point to the sorted table entry currently under consideration. After the final sorted table index has been completed, the array of pointers to the subsequences having the lowest error contains the optimal results of the search. If the length of the sequence has been constrained, all that is necessary is to select the backpointer for the specified length sequence. If not, the second half of the algorithm shown in Figure 9 is used to determine the best sequence out of those specified by the array of pointers to the best sequence.

Although the algorithm described in this subsection was developed as a natural extension to existing sequence finders that did not save optimal subsequences, reference should be made to the work of others on this problem. Most widely published is the work of Jelinek et al. at IBM.[15,17,18] The IBM work was predicated on a phonemic representation of the recognized speech. The descriptions were in terms of probabilistic finite state machines where the recognized phonemes are outputs of state-to-state transitions, with all state transitions having associated *a priori* probabilities.

[17]F. Jelinek, L.R. Bahl, and R.L. Mercer, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," *IEEE Transactions on Information Theory*, IT-21:250–256, May 1975.

[18]L.R. Bahl and F. Jelinek, "Decoding for Channels With Insertions, Deletions, and Substitutions With Applications to Speech Recognition," *IEEE Transactions on Information Theory*, IT-21:404–411, July 1975.

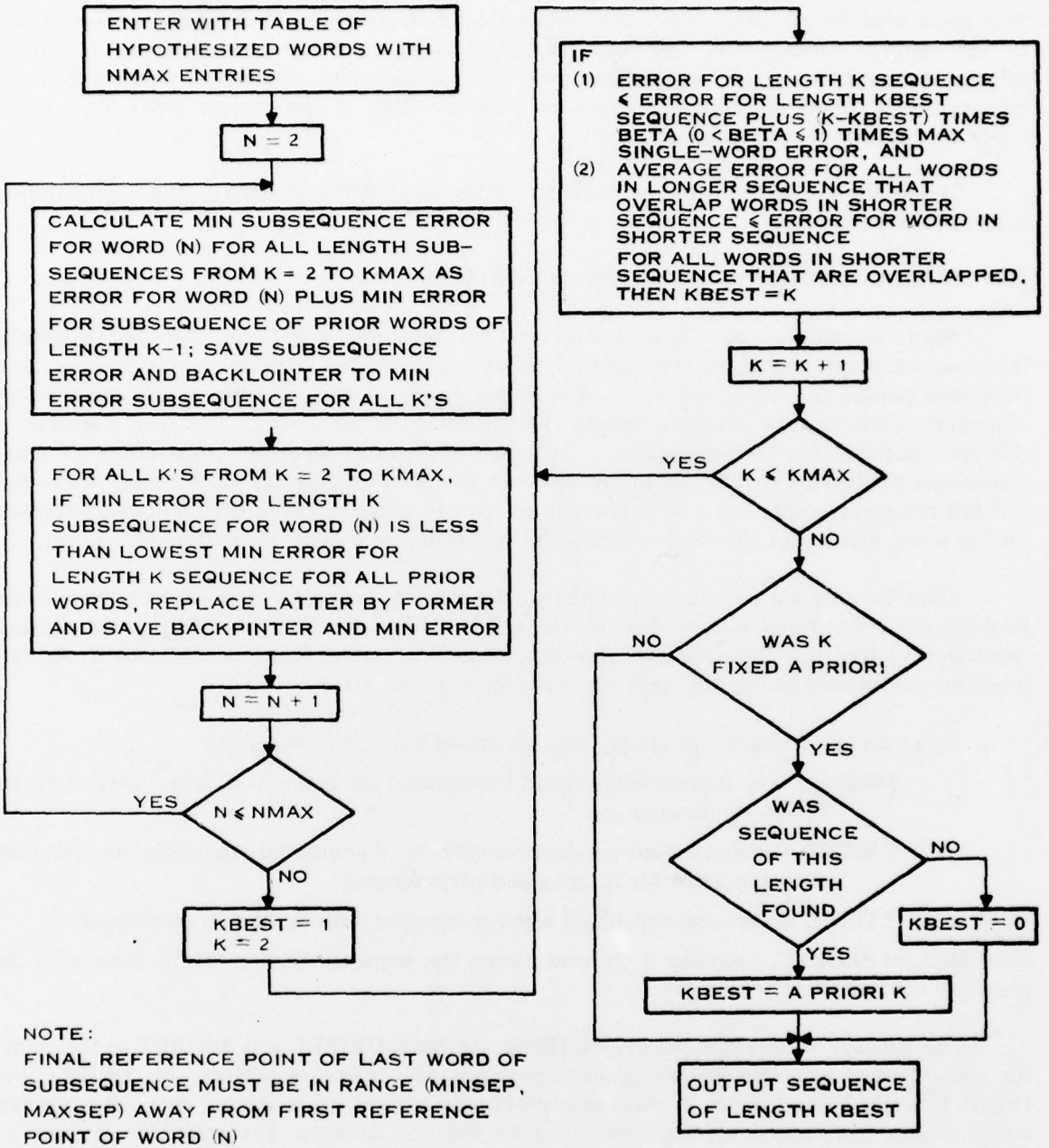


Figure 9. Flow Chart for Efficient Tree-Searching Algorithm

Much closer to the work of this subsection is that of Porter,[12] whose MINT algorithm is used to find the highest probability path through a directed graph of hypothesized words, such as that shown in Figure 8. The nodes (rather than the transitions, as in the IBM work) are each associated with a hypothesized word and the edges represent backpointers to all allowable predecessors. The solution to the real-time processing is almost the same as that described here; each node need be processed only once, rather than each sequence once, by saving backpointers to the optimal subsequence. The only difference is that, in the present study, N optimal subsequences are saved (N is the maximum allowable length sequence), whereas just one backpointer is saved by Porter. Subsequent postprocessing in the present study then selects which of the final N sequences is the best.

The interested reader is referred to Porter[12] for an extended discussion of the probabilistic basis for this procedure.

F. OVERALL WORD-RECOGNITION ALGORITHM

Word recognition at Texas Instruments is currently based on the piecewise-linear time-normalization technique (Subsection II.B) of finding potential acoustic boundaries (reference points) and fitting sequences of reference points together to form hypothesized words. Time-normalized spectral patterns, formatted for the hypothesized words, are then compared to reference patterns (for either speaker-independent or speaker-dependent recognition of either continuous or discrete speech). If the comparison is a good enough match, the time of the first and last reference points and a total normalized error (distance between the input and reference) for the word, along with the label for the word, are stored in a table of hypothesized words.

After the utterance has been completed, the table is sorted by time of occurrence of the final reference point and is then used by the tree-searching algorithm described in the previous subsection to find the best sequence of words. A summary flow chart for the word recognition programs investigated during this contract is shown in Figure 10.

Three specific computer programs were generated during this study:

DIGREC, for speaker-independent recognition of connected digits using only the TI 980 minicomputer

DIGRCT, for speaker-independent recognition of connected digits using the AP 120B array processor for filtering and preprocessing

RTENR, for speaker-dependent word recognition with automatic enrollment.

Note that for DIGREC, sampling is stopped during the sequence finding that is done after the complete utterance has been input.

The primary differences between RTENR and both DIGREC and DIGRCT is the source for the reference scanning and recognition patterns. The reference patterns for DIGREC and DIGRCT were derived from a clustering procedure applied to a design data set collected, digitized and hand-edited off-line before use by the test subjects. The reference patterns for RTENR, however, were derived from on-line enrollment of all the vocabulary words by each subject using the system. These were speaker-specific reference patterns. Both procedures are described in more detail in Sections III and IV.

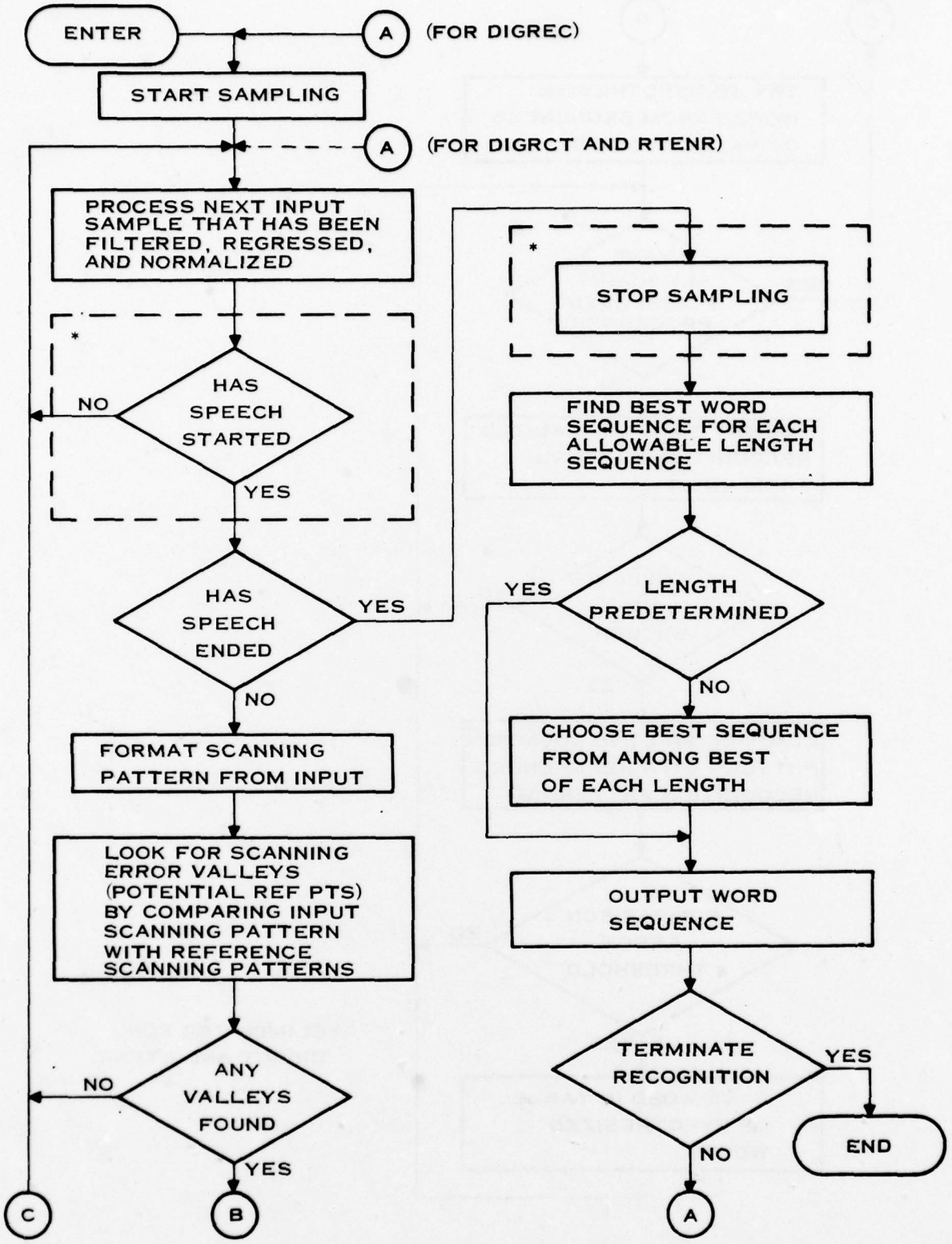


Figure 10. Word Recognition Algorithm Flow Chart (Sheet 1 of 2)

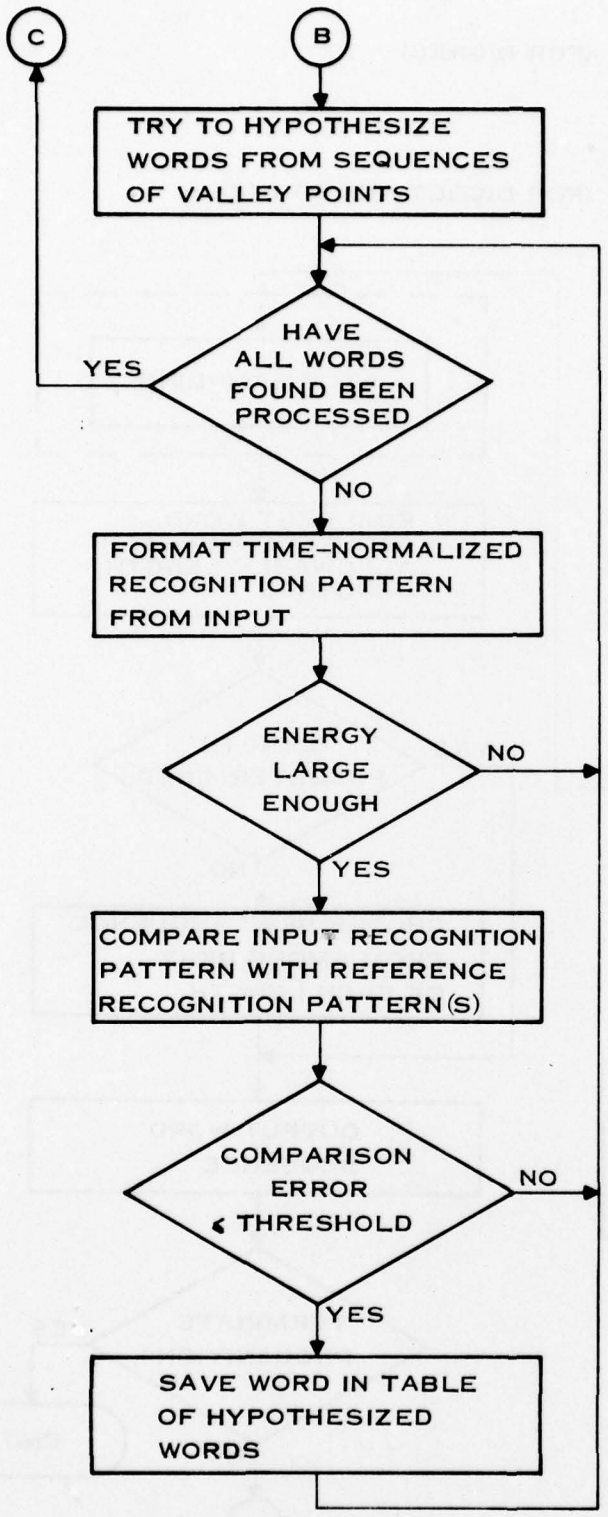


Figure 10. Word Recognition Algorithm Flow Chart (Sheet 2 of 2)

SECTION III
SPEAKER-DEPENDENT WORD RECOGNITION

A. INTRODUCTION

The speech-processing algorithm described in Section II provides the framework for both speaker-dependent and speaker-independent word-recognition tasks. The algorithm is made specific to the given task through definition of the reference scanning and recognition patterns. The speaker-dependent task is accomplished through definition of a single set of reference scanning patterns for each vocabulary word for each speaker. In contrast, multiple scanning patterns are defined for each word in the speaker-independent word-recognition task (discussed in Section IV).

The reference patterns for the speaker-dependent word-recognition task are obtained in a single enrollment session where each word in the vocabulary is spoken in isolation. Intersession variations and contextual variations in continuous speech are accounted for by a method of supervised updating.

The remainder of this section discusses the definition of the reference scanning and recognition patterns, the method of supervised updating, and the application of the algorithm to continuous speech.

B. ENROLLMENT

Enrollment in the speaker-dependent word-recognition task defines the speaker-specific reference patterns for each word in the vocabulary. A total of 20 words per speaker is allowed. Each word is identified to the system and then spoken four times in isolation. These four repetitions are used to define reference scanning and recognition patterns for the word.

The enrollment strategy consists of preprocessing the data for each word, locating reference points, defining scanning patterns, and, finally, defining a recognition pattern. The preprocessing step uses the algorithm defined in Section II to provide the spectrum, energy, regression coefficients, and T-function.

The T-function is a measure of the change in the spectrum, regression coefficients and energy and for time t_j is given by:

$$T_j = \sum_{k=1}^{2} \left[\| (\vec{A}_{j+k})_N - (\vec{A}_{j+k-3})_N \|^2 + \| \vec{C}_{j+k} - \vec{C}_{j+k-3} \|^2 \right.$$
$$\left. + \frac{1}{4} \| E_{j+k} - E_{j+k-3} \|^2 + \frac{1}{4} \| E_{j+k-2} - E_{j+k-4} \|^2 \right]$$

where

$(\vec{A}_j)_N$ = normalized amplitude vector (Appendix A)

$\vec{C}_j$ = regression coefficient vector = $(c_{j1}, c_{j2})^T$

E_j = normalized scanning pattern energy.

Reference points are located in each of the four enrollment words in turn for use in defining scanning patterns and recognition patterns. The steps in locating the reference points are as follows:

(1) Locate the beginning, i_{ST}, and the end, i_{END}, of the word using an energy threshold.

(2) Sum the energy in the word segment from i_{ST} to i_{END} to obtain S_E.

(3) Locate the time points associated with 5, 10, 90, and 95 percent of the energy sum S_E, that is (i_5, i_{10}, i_{90}, i_{95}).

(4) Locate all the T-function peaks in the word segment.

(5) If a T-function peak exists in the interval $[i_{ST}, i_{10}]$, define its location as the first reference point, RP_1; if not, let $RP_1 = i_5$.

(6) If a T-function peak exists in the interval $[i_{90}, i_{END}]$, define it as the last reference point, RP_N; if not, let $RP_N = i_{95}$.

(7) Generate the set, T, of all T-function peaks in the interval (RP_1, RP_N). If T is null, then the word has only the two reference points, RP_1 and RP_N.

(8) If T is not null, use the elements of T in all combinations to maximize the function:

$$F = \prod_{k=1}^{N} \left(\frac{T_k}{T_{min}} \right) \prod_{k=2}^{N} \left(\frac{i_k - i_{k-1}}{i_{RP_N} - i_{RP_1}} \right)^r$$

where $i_1 = RP_1$; $i_N = RP_N$; $i_k \epsilon T$ for $k = 2 \ldots, N - 1$; T_k is the value of the T-function at i_k; T_{min} is a normalization factor; and r is the power for the distance weighting. The subset of T that maximizes the function F is then used as the set of reference points for the word, with the first reference point being RP_1 and the last being RP_N. The objective of the maximization is to distribute the reference points uniformly throughout the word.

At the location of each of the reference points thus defined, a scanning pattern is defined as discussed in Section II. The scanning pattern uses the spectrum, energy and regression coefficients, and their respective differences between time samples.

The definition of the recognition patterns also makes use of the location of the reference points. The steps in defining the recognition pattern format are as follows:

(1) If the energy is greater than a threshold for time samples $i_{RP_1} - 4$, $i_{RP_1} - 3$, $i_{RP_1} - 2$, and $i_{RP_1} - 1$, then extrapolation columns are defined at $i_{RP_1} - 4$ and $i_{RP_1} - 2$.

(2) If the energy is greater than a threshold for time samples $i_{RP_N} + 1$, $i_{RP_N} + 2$, $i_{RP_N} + 3$, and $i_{RP_N} + 4$, then extrapolation columns are defined at $i_{RP_N} + 2$ and $i_{RP_N} + 4$.

(3) Interior to every pair of reference points, i_k and i_{k+1}, interpolate with M columns for the recognition pattern, where

$$M = \begin{cases} 2 & i_{k+1} - i_k < 4 \\ 3 & 4 \leq i_{k+1} - i_k \leq 10 \\ 5 & 10 < i_{k+1} - i_k \leq 20 \\ 8 & 20 < i_{k+1} - i_k \end{cases}$$

The format just described is used to define a recognition pattern using the procedure outlined in Section II.

Once the reference points, scanning patterns, and recognition pattern have been obtained for one of the four repetitions of a word as described above, those scanning patterns are used to scan the remaining three repetitions to automatically find reference points and define reference patterns. At each repetition, the new scanning and recognition patterns are averaged with the old patterns. Each of the four repetitions of the word are used in turn to define a set of reference scanning and recognition points. As each reference pattern is formed, a composite error consisting of the scanning error and the recognition error is computed. The minimum composite error over the four different enrollment trials defines the ultimate enrollment for the word.

An example of automatic enrollment is shown in Figure 11 for the word "Two". For an energy threshold of 100, the beginning of the word, i_{ST}, is at 26 and the end, i_{END}, is at 66. The 5-, 10-, 90-, and 95-percent energy sum points are at 29, 33, 55, and 58, respectively. Therefore, the strategy outlined above locates the reference points at 26, 33, and 58. The recognition format consists of three interpolation points between the reference points at 26 and 33, eight interpolation points between the reference points at 33 and 58, and the extrapolation beyond the reference point at 58 by two and four points.

C. UPDATING

To accommodate intersession variations and continuous-speech context variation, several sessions of supervised updating should be performed. The updating should consist of five sessions separated by at least a day. A series of phrases that contain all the transitions for the 20-word vocabulary should be spoken continuously. If the phrase is recognized, the reference patterns are updated by adding 1/16 of the new pattern to 15/16 of the old patterns. The five sessions spaced at 1-day intervals adapts the reference pattern for intersession variations. The series of phrases with different contexts allows the reference patterns to adapt to continuous speech by allowing the patterns to "see" something besides silence between the words and also to account for coarticulation which occurs in some contexts.

D. *APPLICATION TO CONTINUOUS SPEECH*

The method of word recognition using spectral pattern matching offers a dramatic improvement in performance compared with schemes that rely on finding word boundaries with energy profiles. The spectral pattern-matching method works well in continuous speech provided the words are enrolled properly and several sessions of continuous speech updating are accomplished, as discussed in Subsection III.C. The example in Figure 11 of a good enrollment for the word "two" shows the reference points. In the example, the registration points were interior to the word, so that the scanning patterns will not be confused with the scanning

patterns of adjacent words. With several sessions of continuous updating using phrases containing all the word transitions, the scanning and recognition patterns adapt to "see" all these transitions.

Unfortunately, the method of automatic enrollment described above does not always give a good enrollment. As an example of a poor enrollent, Figure 12 shows the automatic enrollment for the word "six". For an energy threshold of 100, the beginning of the word, i_{ST}, is at 25 and the end, i_{END}, is at 44. The 5-, 10-, 90-, and 95-percent energy sums are shown at 30, 31, 39, and 41, respectively. The automatic enrollment scheme chose the reference points at 29 and 44. The recognition pattern consists of five columns between the two reference points. There should have been another reference point at 56 and extrapolation of the recognition pattern both before the first reference point at 29 and after the last reference point at 56. As the patterns exist with the automatic enrollment, the updating will not improve the recognition of the word "six".

It is believed that an improved automatic enrollment algorithm would consist of a set of speaker-independent reference phoneme patterns. Given the phonetic spelling, the specific phoneme patterns for the word being enrolled would be scanned across the input data for that word and scanning errors obtained. The minimum scanning errors would be located and used in a dynamic programming algorithm to obtain the best sequence of phonemes in the proper spelling order for the word. T-function peaks between the minimum error locations for the phoneme pairs would be used to define the reference points for the word. The recognition pattern format would be specific to the phonemic spelling of the word. Once the reference points and recognition pattern format are defined, the enrollment procedure would be the same as defined above.

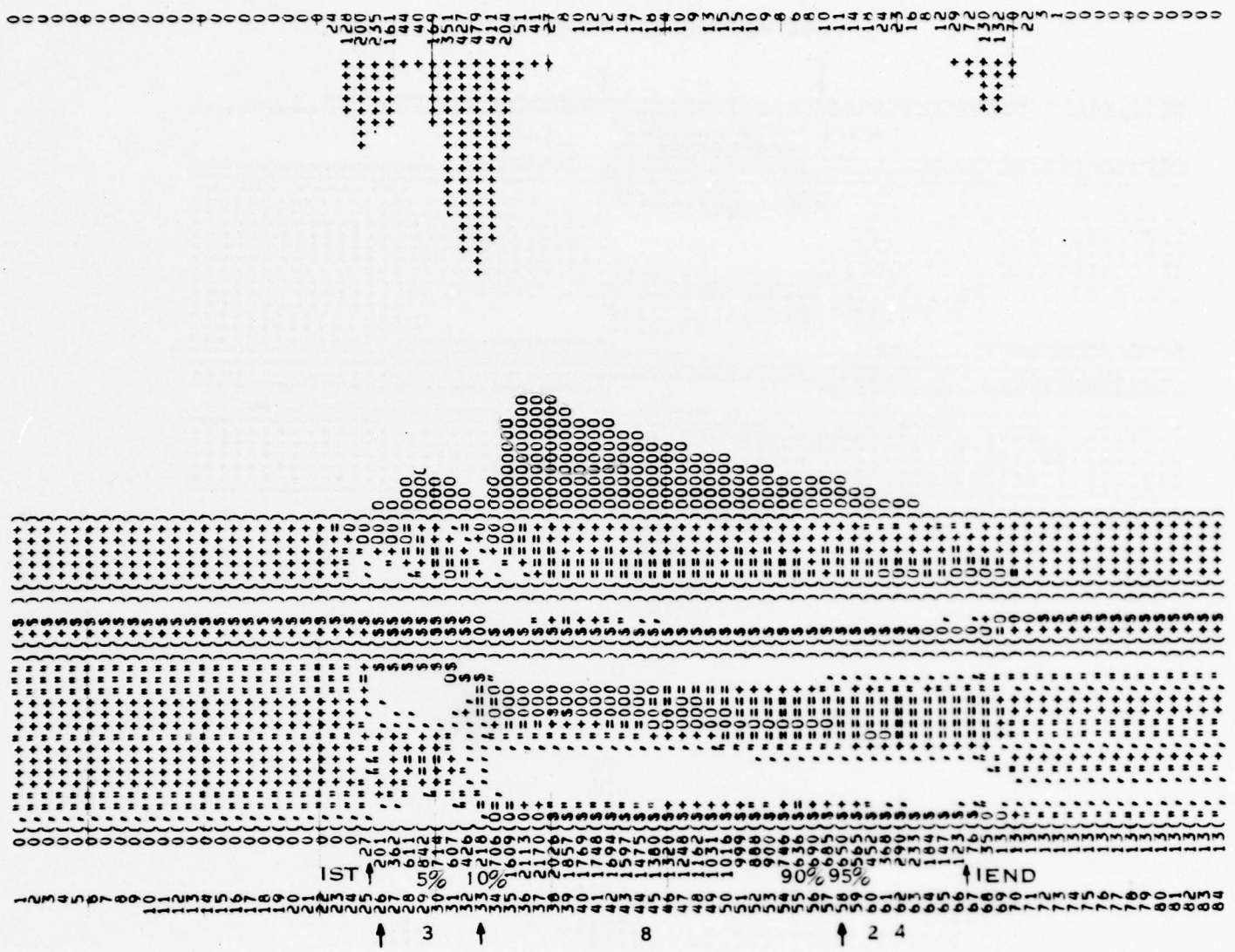


Figure 11. Automatic Enrollment for "Two"

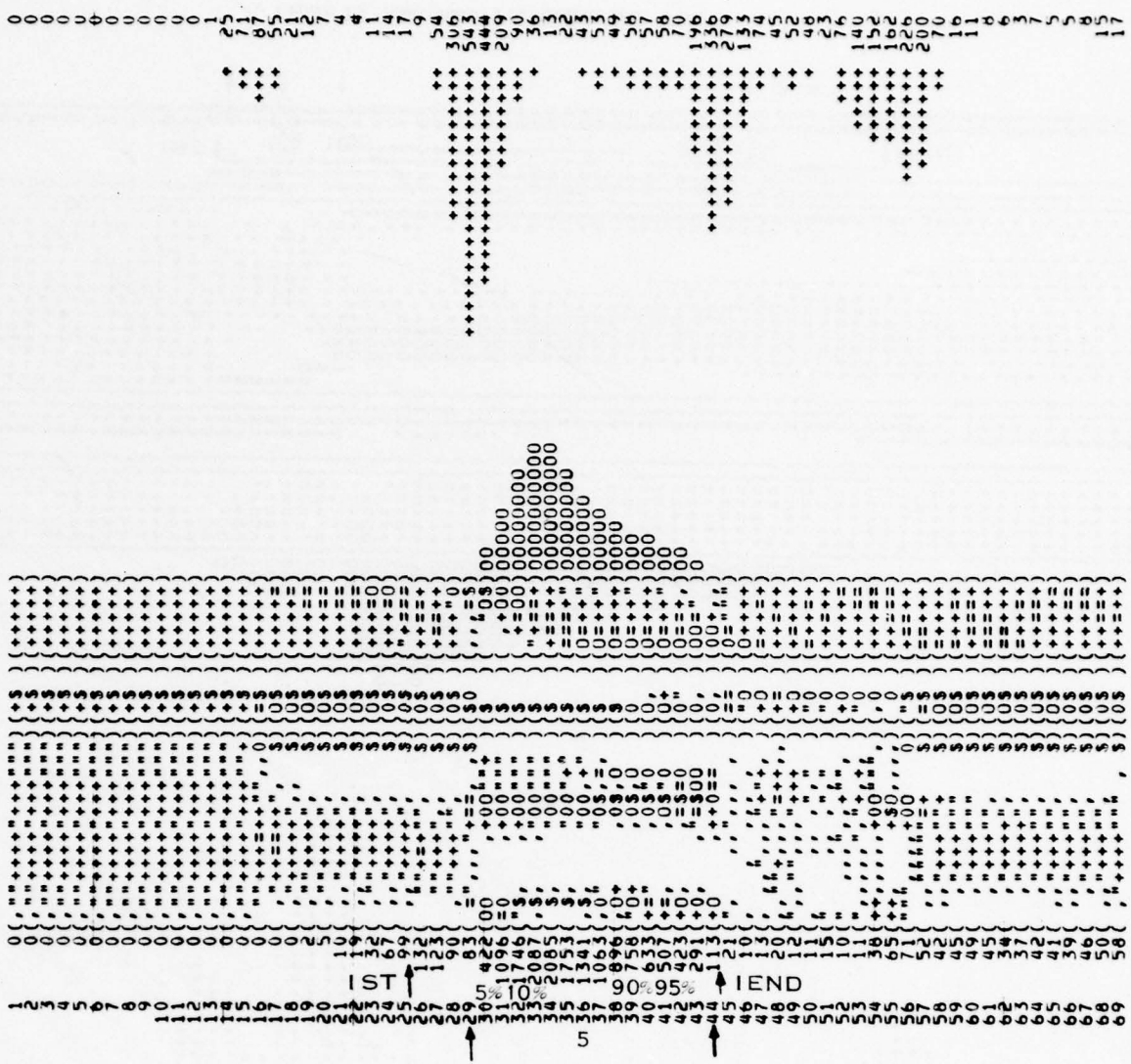


Figure 12. Automatic Enrollment for "Six"

SECTION IV

REFERENCE-PATTERN GENERATION FOR SPEAKER-INDEPENDENT WORD RECOGNITION

The creation of reference patterns for speaker-dependent, isolated word recognition is fairly straightforward: extract patterns from a single enrollment session and accommodate intersession variation through updating of the reference patterns (learning with a teacher). For speaker-independent word recognition, however, the increased variance of input data from singe-reference templates because of dialect, idiolect, and actual physical characteristics of the speaker (length and shape of the vocal tract, pitch, etc.) requires a more complex approach. Of course, allowing continuous speech input exacerbates the problem with the introduction of contextual variations. An obvious solution is to allow multiple reference templates for each word. One approach to deriving a set of multiple reference templates from a design data set is to partition the data set on the basis of information other than the actual data, such as sex or linguistic background. A second approach, the one used in the total voice verification study and in the present study, is to partition the data on the basis of the data points themselves using clustering techniques. The remainder of this section reviews the use of clustering in speaker-independent reference template generation, discusses the clustering used in the studies at Texas Instruments, and gives results of some further analysis of the patterns developed during the total voice work, patterns that were also used on the current unconstrained digit recognition work in order to preserve compatibility.

A. REVIEW OF CLUSTERING IN SPEAKER-INDEPENDENT WORD RECOGNITION

Except for the work done in this study and in the total voice study,[1] the only other applications of clustering to speaker-independent reference-pattern generation has been concurrent work started independently about the same time at Bell Laboratories (as reported by Rabiner, Levinson, Rosenburg, and Wilpon[19-22]) and subsequent independent work done in Japan by Tanaka.[23,24] The application at Bell Laboratories is isolated word recognition, and Tanaka's procedure has been applied only to the recognition of stop consonants. The remainder of this subsection reviews these other works.

[19] L.R. Rabiner, "On Creating Reference Templates for Speaker-Independent Recognition of Isolated Words," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-26:34–42, February 1978.

[20] S.E. Levinson et al., "Interactive Clustering Techniques for Selecting Speaker-Independent Reference Templates for Isolated Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-27:134–141, April 1979.

[21] L.R. Rabiner et al., "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Washington, D.C., 574–577, 2–4 April, 1979.

[22] L.R. Rabiner and J.G. Wilpon, "Considerations in Applying Clustering Techniques to Speaker-Independent Word Recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, Washington, D.C., 578–581, 2-4 April, 1979.

[23] K. Tanaka, "A Standard Category Pattern-Making Method With Application to Phoneme Recognition," *Proceedings of the Fourth International Joint Conference on Pattern Recognition*, Kyoto, Japan, 1030–1032, 7-10 November, 1978.

[24] K. Tanaka, "A Talker Clustering Method for Standard Pattern Making," *Progress Report on Speech Research '77*, Electrotechnical Laboratory, Japan, August 1978.

Although the Bell Laboratories work has been for words said in isolation, the word sets investigated, while including the digits (0 through 9), were considerably larger. One set was a 54-word vocabulary proposed originally by Gold,[25] and the other set contained the alphabet, the digits, and three control words. The speech representation chosen was a set of linear predictive coding (LPC) parameters for each 15-ms frame of speech. These parameters then underwent a time normalization using a dynamic programming technique.[26] The similarity measure used in the Bell Laboratories work was one of the following form proposed by Itakura:[7]

$$d[k, w(k)] = \log \frac{\vec{a}_{w(k)} \, V \, \vec{a}_{w(k)}}{\vec{a}_k \, V \, \vec{a}_k} \tag{5}$$

where $\vec{a}_k$ is the vector of LPC coefficients associated with the k*th* frame of the test or unknown utterance x_i; $a_{w(k)}$ is the vector of LPC coefficients derived from the w(k)*th* frame of the reference utterance x_j; and V is the matrix of autocorrelation coefficients computed from the k*th* frame of the test utterance. Note that this distance measure is not a true metric since it is not symmetrical.

The clustering technique reported in Levinson et al.,[20] and Rabiner et al.,[21] is a supervised, interactive procedure and is the combination (figure 13) of the following four procedures: chainmap, shared nearest neighbor, k-means, and a version of ISODATA. The details of their procedures are given in Levinson et al.[20] In this approach, the investigators first attempted to find good estimates of both the number of clusters (using the chainmap) and their cluster centers (using the k-means) for input to an iterative optimization procedure (ISODATA) that allowed splitting and merging of clusters. The overall intent was to maximize a quality measure σ for the assignment of N observations into M classes. The value of σ is given by

$$\sigma = \frac{\dfrac{1}{M(M-1)} \displaystyle\sum_{i=1}^{M} \sum_{j=1}^{M} \delta\left(x_p^{(i)}, x_p^{(j)}\right)}{\dfrac{1}{M} \displaystyle\sum_{i=1}^{M} \frac{1}{m_i(m_i-1)} \sum_{j=1}^{m_i} \sum_{k=1}^{m_i} \delta\left(x_j^{(i)}, x_k^{(i)}\right)} \tag{6}$$

where superscripts indicate class membership, p subscripts indicate reference-class prototypes, and $\delta(a,b)$ is a nonsymmetric similarity measure between patterns a and b that is the average of the Itakura distances over all the frames of the reference pattern. Further comments regarding σ are made later in this section.

[25] B. Gold, "Word-Recognition Computer Program," Massachusetts Institute of Technology, Cambridge, RLE Technical Report 452, June 1966.

[26] L.R. Rabiner et al., "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP-26:575–582, December 1978.

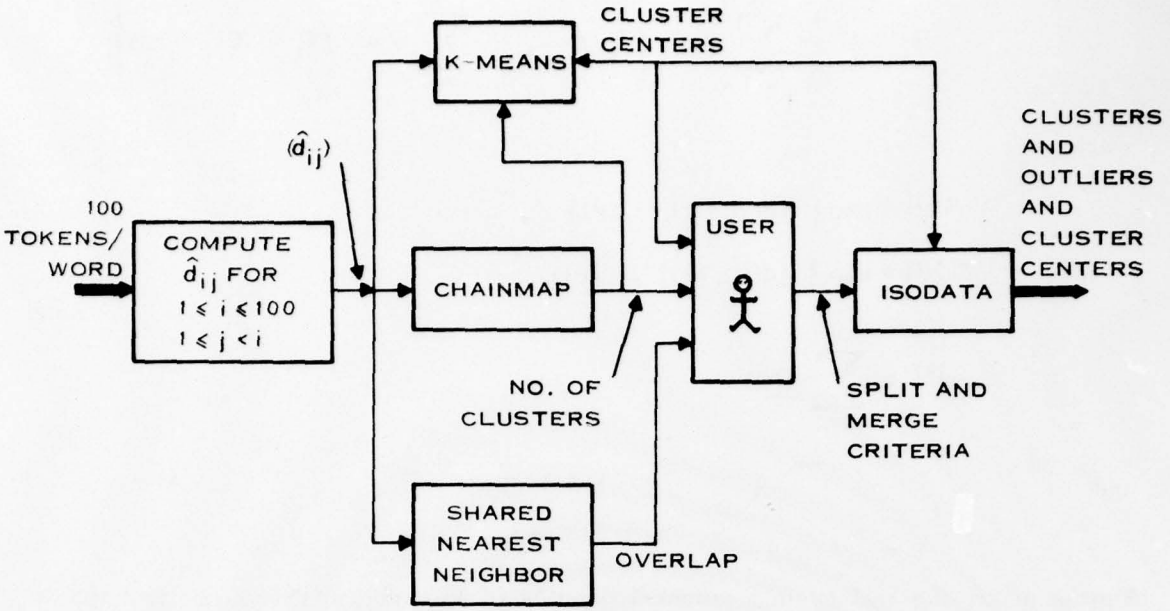


Figure 13. Bell Laboratories Clustering Procedures (From Rabiner, et al[21])

Speaker-independent recognition results for isolated digits given in Rabiner et al.[21] range from 97.5 to 100 percent. Results for the entire 39-word set range from about 50 to 80 percent, and recognition improves with the number of reference templates used for each word.

Rabiner and Wilpon[22] extended the previous work to unsupervised clustering using the same data set, distance, and quality measure (σ) as previously used. One clustering algorithm uses only precomputed distances between observations, attempting to place each observation uniquely in a cluster with all others that are similar, and a second clustering algorithm combines (by averaging) observations that are similar. Comparisons were made in this work among three different LPC feature sets and between cluster representation either by the data point with the minimum maximum distance from all other points in the cluster or by the average of all the points in the cluster. The results of Rabiner and Wilpon indicate that the algorithm using precomputed distances was superior to the other and that the use of an averaged pattern to represent the cluster was superior to using the minimum maximum center. Again, the recognition accuracy improved with the number of reference templates used.

Tanaka[23,24] clusters a set of observations into different classes by moving each observation iteratively by some amount proportional to the density of points in the neighborhood of the observation, where each point is modified by its gradient with respect to all other points. Specifically, for the set of observation vectors x_i ($i = 1, \ldots, N$) for the *jth* iteration,

$$\vec{x}_i^{j+1} = \frac{1}{w_i^j} \sum_{k=1}^{N} \delta(s^j) \left[\vec{x}_k^j + \frac{c}{\sqrt{w_k^j}} \sum_{r=1}^{N} \sqrt{w_r^j} \left(\vec{x}_k^j - \vec{x}_r^j \right) \delta(2s^j) \right] \quad (7)$$

where

$$\delta(s^j) = \exp\left\{ -[d(x_i^j, x_k^j)]^2 / 2(s^j) \right\}$$

$$\delta(2s^j) = \exp\left\{ -[d(x_k^j, x_r^j)]^2 / 2(2s^j) \right\}$$

$$w_i^{j+1} = \sum_{k=1}^{N} \delta(s^j)$$

and

$$s^j = s^{j-1} / 3\sqrt{2}$$

Tanaka makes the analogy to a potential function of an exponential form, so that the $(x^j_k - x^j_r)\ \delta(2s^j)$ term can be considered a gradient of the potential function. Hence, the term in brackets in Equation (7) represents one of the points modified by the gradient of a potential function of that point with respect to all other points. These modified points are then used in N weighted sums to determine each of the new N points for the (j + 1)*st* iteration. Clustering stops when the window $\delta(s^j)$ has narrowed sufficiently that every data point is the same as that of the previous stage. The number of iterations and the final number of clusters are obviously affected by the choice of c and s^0.

This approach is quite similar to that presented by Fukunaga and Hostetler.[27] However, they essentially use the points as modified in the brackets in Equation (7) as the new N points for the (j + 1)*st* iteration, instead of using weighted sums of such points.

Tanaka applied this method to generation of reference patterns for use in the difficult problem of detecting the three stop consonants /p,t,k/. This effort is directed toward a phonetic classification-based speech recognition system. Tanaka's results are 89 percent for the test data and 82 percent for stop consonants of other speakers.

The method used in the study covered by this final report and the method used in the preceding total voice study differ significantly from the approaches just described. Tanaka's approach differs not only in terms of using phonemic-based recognition but also in the clustering by allowing movement of the data points. Although differing final clusters and numbers of clusters could be produced in Tanaka's algorithm by varying the parameters, he does not discuss how to choose the final clusters. Differing applications do not allow comparison of his final results to those presented here.

[27]K. Fukunaga and L.D. Hostetler, "The Estimation of the Gradient of a Density Function, With Applications in Pattern Recognition," *IEEE Transactions on Information Theory*, IT-21:32–40, January 1975.

Although the application and the clustering approach used at Bell Laboratories differ less, their approach is to find a single "best" set of clusters, ultimately using an ISODATA algorithm that can split and merge clusters. The approach used at Texas Instruments, however, finds good estimates for cluster centers using a hierarchical clustering approach, performing an iterative optimization on cluster definitions for several fixed values of M (later referred to as "c"), and choosing M based not only on criterion values for each of the final partitions, but also on a subjective evaluation of the final cluster averaged patterns themselves.

A criterion similar to that used by Bell Laboratories (σ) is compared with the one used here (trace of the within-class scatter matrix) later in this section. In addition, although a common data base could not be used, a small test of recognition performance on isolated digits was performed to provide a rough comparison with the isolated digit results presented by Rabiner et al.[21]

B. DETAILED CLUSTERING ALGORITHM

The clustering algorithm used in the total voice study and extended in the present study represents a unique combination of several methods, all centered on the use of Euclidian distances because the fast vector comparator exists peripheral to the TI 980 to perform the computation. The entire procedure is shown in Figure 14. The patterns used in the speaker-independent digit-recognition evaluations were generated during the total voice study using the path through the procedure denoted by the double lines in Figure 14. The other paths in Figure 14 were added during this study for evaluation of and consistency checks on the previous patterns and for rudimentary outlier analysis.

A detailed description of the procedure used to generate the patterns and the patterns themselves are given in the total voice final report.[1] A brief description of the procedure is given here for completeness. The method used in the total voice study to derive the patterns used in the evaluation was an agglomerative method combining the two clusters that have the smallest average distance (MINAVE) between the points in the two clusters, i.e., combining the i and j clusters that have the minimum

$$\frac{1}{n_i n_j} \sum_{\vec{x} \epsilon \chi_i} \sum_{\vec{x}' \epsilon \chi_j} d(\vec{x}, \vec{x}') \tag{8}$$

where

n_i = number of $\vec{x}$ in class χ_i

n_j = number of $\vec{x}'$ in class χ_j

and, in this case,

$$d(\vec{x}, \vec{x}') = ||x - x'||^2$$

The second step used was to improve on the partitions from the hierarchical clustering iteratively by moving samples from one group to another if such a move improved the value of

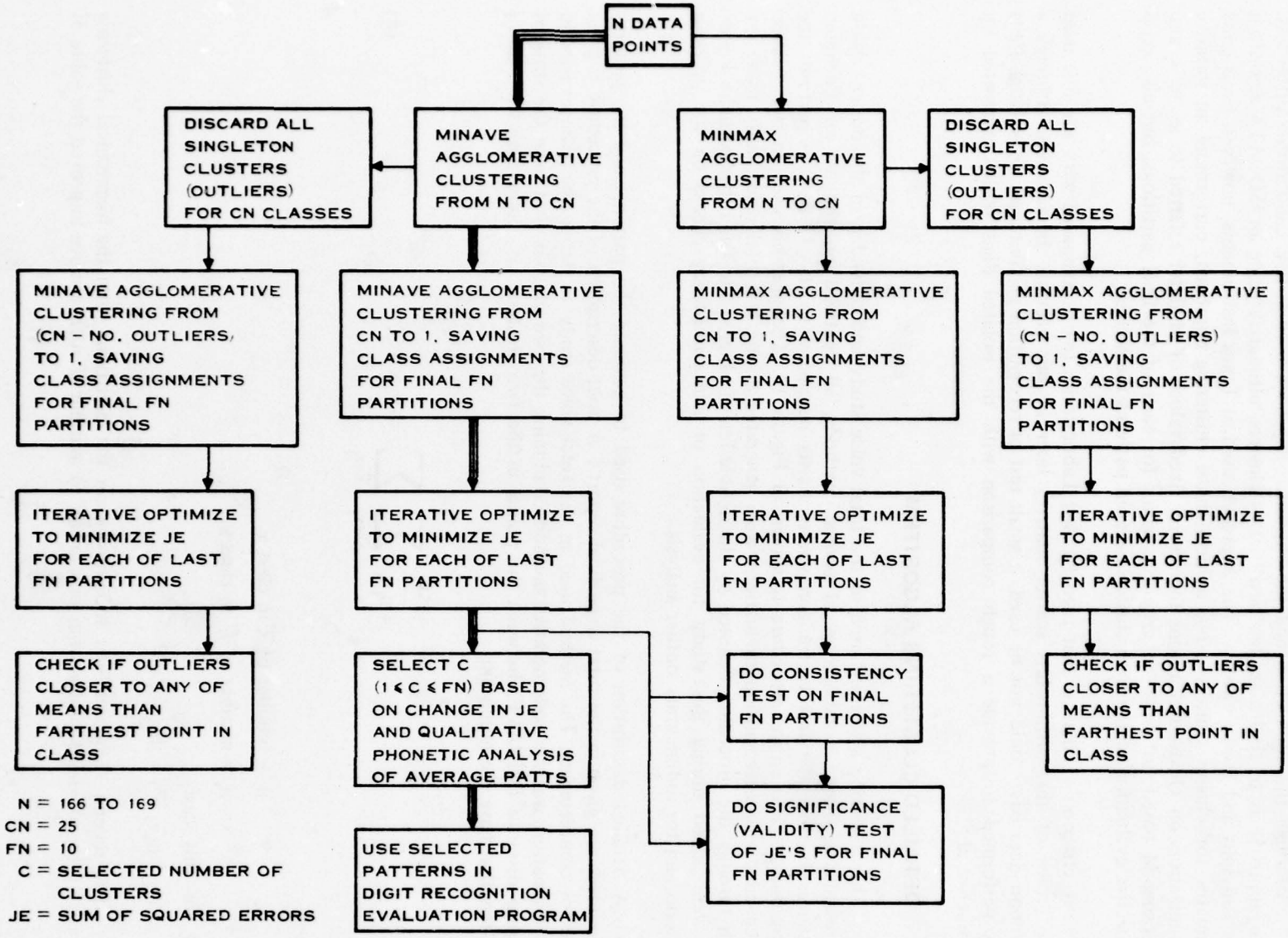


Figure 14. Block Diagram of Clustering Procedure Developed for Clustering Scanning and Recognition Patterns for Speaker-Independent Digit Recognition

some criterion function. This step used the iterative optimization method of Duda and Hart[28] that minimized the sum-of-squared error criterion J_e, written as

$$J_e = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{\hat{x} \epsilon \chi_i} \| \hat{x} - \vec{m}_i \|^2 \tag{9}$$

where

$$\vec{m}_i = \frac{1}{n_i} \sum_{\hat{x} \epsilon \chi_i} \hat{x}$$

If a point $\hat{x}$ is moved from class χ_i to class χ_j, the means $\vec{m}_i$ and $\vec{m}_j$ change to

$$\vec{m}_i^* = \vec{m}_i - \frac{\hat{x} - \vec{m}_i}{n_i - 1} \quad \text{and} \quad \vec{m}_j^* = \vec{m}_j + \frac{\hat{x} - \vec{m}_j}{n_j + 1} \tag{10}$$

The value of J_i decreases to

$$J_i^* = J_i - \frac{n_i}{n_i - 1} \| \hat{x} - \vec{m}_i \|^2 \tag{11}$$

and J_j increases to

$$J_j^* = J_j + \frac{n_j}{n_j + 1} \| \hat{x} - \vec{m}_j \|^2 \tag{12}$$

Clearly, then, since the criterion is to minimize J_e, if

$$\frac{n_j}{n_j + 1} \| \hat{x} - \vec{m}_j \|^2 < \frac{n_i}{n_i - 1} \| \hat{x} - \vec{m}_i \|^2 \tag{13}$$

then $\hat{x}$ should be transferred from class χ_i to class χ_j. Specifically the point $\hat{x}$ is moved to the class χ_j, having the smallest $(n_j/n_j + 1) \| \hat{x} - m_j \|^2$.

An additional property (not necessarily good) of the selection of J_e as a criterion is that a set of equally divided clusters is favored over a set containing both small and large clusters, as noted earlier. This can be seen by considering $n_i \gg n_j$ in Equation (13), which yields approximately

$$\frac{n_j}{n_j + 1} \| \hat{x} - \vec{m}_j \|^2 < \| \hat{x} - \vec{m}_i \|^2 \tag{14}$$

[28] R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis*, John Wiley and Sons (New York, 1973).

Thus, for $n_j = 1$, the distance $||\hat{\vec{x}} - m_j||^2$ need only be less than twice the distance $||\hat{\vec{x}} - m_i||^2$ to the old mean for $\hat{\vec{x}}$ to the transferred to class χ_j.

C. CRITERIA FOR MEASURING PARTITION GOODNESS

Although minimization of J_e was the criterion used in the iterative optimization and $(J_e^c - J_e^{c+1})/J_e^c$ was used as a second criterion to aid in selecting the number of clusters, the values of several other criteria related to J_e were calculated during the current study for all patterns for numbers of classes from 1 to FN (= 10). (Superscripts on J_e are used to denote number of classes.)

The discussion in the remainder of this subsection assumes a knowledge of scatter matrices. Appendix B has been provided for those not familiar with the concept.

A third criterion is the value of tr S_B/tr S_W, which is inherently maximized during the iterative optimization by the minimization of J_e (= tr S_W). Note that since tr $S_T = J_e^1$ and tr S_B = tr S_T − tr S_W, then tr S_B/tr $S_W = (J_e^i - J_e^c)/J_e^c$ for c classes.

A fourth related criterion suggested by Hartigan[29] for choosing the number of clusters is $(n - c)(J_e^c - J_e^{c+1})/J_e^c$. Hartigan suggests that values of this ratio greater than 10 justify increasing the number of clusters.

A fifth criterion is related to the F-ratio from analysis of variance, taking into account the degrees of freedom of tr S_B and tr S_W. This criterion is given by

$$\frac{\text{tr } S_B/(c-1)}{\text{tr } S_W/(n-c)} = \frac{(n-c)(J_e^1 - J_e^c)}{(c-1)\,J_e^c} \tag{15}$$

and is attribted by Everitt[30] to Calinski and Harabasz.[31]

The sixth criterion calculated during this study is a σ analogous to that used in the Bell Laboratories studies.[19-22] The value of σ is calculated by

$$\sigma = \frac{\dfrac{1}{c(c-1)} \displaystyle\sum_{i=1}^{c}\sum_{j=1}^{c} \|\vec{m}_i - \vec{m}_j\|^2}{\dfrac{1}{c} \displaystyle\sum_{i=1}^{c} \frac{1}{(n_i - 1)} \left[\frac{1}{n_i} \sum_{\vec{x}\epsilon\chi_i} \sum_{\vec{x}'\epsilon\chi_i} \|\vec{x} - \vec{x}'\|^2 \right]} \tag{16}$$

[29] J.A. Hartigan, *Clustering Algorithms*, John Wiley and Sons (New York, 1975).
[30] B. Everitt, *Cluster Analysis*, Heinemann Educational Books, Ltd., (London, 1974).
[31] T. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis," (unpublished), 1971.

The relationship between σ and the tr S_B/tr S_W criterion can be seen better by putting both σ and criterion 5 in equivalent forms. The σ term can be rewritten as

$$\sigma = \frac{\dfrac{c}{c-1}\displaystyle\sum_{i=1}^{c}\sum_{j=1}^{c}\frac{1}{c^2}\|\vec{m}_i - \vec{m}_j\|^2}{\dfrac{2}{c}\displaystyle\sum_{i=1}^{c}\frac{1}{(n_i - 1)}J_i} \tag{17}$$

and the fifth criterion multiplied by the factor c/(2n) can be rewritten (see Appendix C) as a seventh criterion α, as follows:

$$\alpha = \frac{c(\text{tr } S_B)/(c-1)}{2n(\text{tr } S_W)/(n-c)} = \frac{\dfrac{c}{c-1}\displaystyle\sum_{i=1}^{c}\sum_{j=1}^{c}\frac{n_i n_j}{n^2}\|\vec{m}_i - \vec{m}_j\|^2}{\dfrac{2}{c}\displaystyle\sum_{i=1}^{c}\frac{1}{\left(\frac{n}{c}-1\right)}J_i} \tag{18}$$

The values of all seven of these criteria for several classes of one of the 34 pattern types clustered in this study are shown in Table 5. The desire is to maximize the last five of the seven criteria discussed above. Note, however, that the sixth criterion, σ, is actually only monitored, while the optimization is on the basis of J_e, from which all the other criteria (except σ) are derived.

TABLE 5. VALUES OF SEVEN CLUSTERING CRITERIA FOR POSTITERATIVE OPTIMIZATION OF MINMAX AGGLOMERATIVE CLUSTERS FOR RECOGNITION PATTERN FOR DIGIT "SIX"

	Criterion						
C	1	2	3	4	5	6	7
1	58,412.0	0.176	0.000	28.850	0.000	0.000	0.000
2	48,136.5	0.051	0.213	8.296	34.795	0.422	0.208
3	45,686.7	0.055	0.279	8.897	22.561	0.426	0.203
4	43,177.7	0.022	0.353	3.473	18.935	0.518	0.227
5	42,246.4	0.030	0.383	4.771	15.306	0.530	0.229
6	40,986.8	0.019	0.425	2.943	13.520	0.486	0.243
7	40,228.1	0.036	0.452	5.652	11.903	0.546	0.249
8	38,789.0	0.003	0.506	0.543	11.346	0.601	0.272
9	38,654.9	0.021	0.511	3.343	9.987	0.601	0.269
10	37,826.5	–	0.544	–	9.372	0.633	0.281

D. DESCRIPTION OF CLUSTER ANALYSIS DOCUMENTATION

This subsection gives a brief description of the printouts available from an analysis program that was run on the output data from the entire clustering procedure shown in Figure 9 for each of the 34 pattern types. A subset of the available outputs is presented in Appendix D. Samples of available printouts are given in this subsection for the scanning pattern for reference-point 1 for the digit zero. MINAVE is used to refer to agglomerative clustering by combining the two clusters having the minimum average distance between all points in the two clusters. Corespondingly, MINMAX refers to agglomerative clustering by combining the two clusters having the minimum maximum distance between the two clusters.

1. Trees

The first type of output available is a tree (dendogram) showing the final joinings or agglomerations in the hierarchical procedures and is available for all four branches of the algorithm shown in Figure 9. Accompanying each dendogram is a table showing the values of the joining criterion for each level and the relationship of the criterion values to the dendogram. The tree-printing subroutine was adapted from appendix G of Anderberg.[32] The tree for the MINAVE hierarchical clustering using all samples is shown in Figure 15 for the joining criteria values in Table 6.

2. Parameter Comparisons

The second type of output from the analysis program gives the values of the six criteria described in Subsection IV.C, the values of the errors during the agglomerative clustering, and the number of iterations required in the iterative optimization to reach the final partitions. The conditions for each of the parameter comparisons produced and a reference to the figure showing an example of that comparison are listed below:

MINAVE, all points; pre- to postiterative optimization comparison (Figure 16)

MINAVE, outliers discarded; pre- to postiterative comparison (Figure 17)

MINMAX, all points; pre- to postiterative optimization comparison (Figure 18)

MINMAX, outliers discarded; pre- to postiterative comparison (Figure 19)

Preiterative optimization, all points; MINAVE to MINMAX comparison (Figure 20)

Postiterative optimization, all points; MINAVE to MINMAX comparison (Figure 21).

3. Consistency Tests

The class assignments obtained using the MINAVE and the MINMAX hierarchical clustering procedures after iterative optimization are compared for the number of clusters ranging from 2 to FN (= 10). This comparison is in terms of two contingency matrices such as shown in Figure 22 for 10 classes after iterative optimization. This output first lists the members of each class for the iteratively optimized results of the MINAVE agglomerative clustering followed by those from the MINMAX agglomerative clustering. The first contingency table then compares the

[32] M.R. Anderberg, *Cluster Analysis for Applications*, Academic Press (New York, 1973).

TABLE 6. CRITERION VALUES FOR TREE FOR FINAL 24 STAGES OF MINAVE AGGLOMERATIVE CLUSTERING OF ALL (166) SCANNING PATTERNS FOR FIRST REFERENCE POINT OF DIGIT "ZERO"

	Class		Criterion	
Stage	I	J	Absolute	Relative
1	16	22	403.692	1
2	7	23	406.000	1
3	2	6	408.710	1
4	7	15	414.500	1
5	10	19	414.750	1
6	9	21	418.000	2
7	1	11	419.967	2
8	9	18	429.875	3
9	13	20	439.000	3
10	2	3	441.115	3
11	4	8	461.250	5
12	14	16	462.714	5
13	1	12	464.742	5
14	1	2	488.888	7
15	1	5	509.195	9
16	1	10	527.124	10
17	4	24	530.000	10
18	1	14	538.098	11
19	4	25	542.947	11
20	1	13	573.328	14
21	1	9	626.332	18
22	1	7	655.048	20
23	4	17	664.150	20
24	1	4	729.398	25

members of the classes from each partition, with each entry in the table showing the number of points that are members of both classes. The second contingency table compares data points in pairs for joint membership or lack of joint membership in the same class. In particular, if two data items are in different classes in a partition, this fact is denoted by a 1 in row or column 1. Otherwise, a 1 appears in row or column 2. Hence, the (2,2) entry in the contingency table indicates how many pairs of the $N(N-1)/2$ pairs of points are in the same class for both partitions, and the (1,1) entry indicates how many of the pairs are in different classes for both partitions.

Ideally, for both contingency tables, all off-diagonal elements will be 0. Hence, a measure of the closeness of the two partitions in both cases is the sum of the diagonal entries divided by the sum of all the entries in the table [N for the first table and $N(N-1)/2$ for the second table].

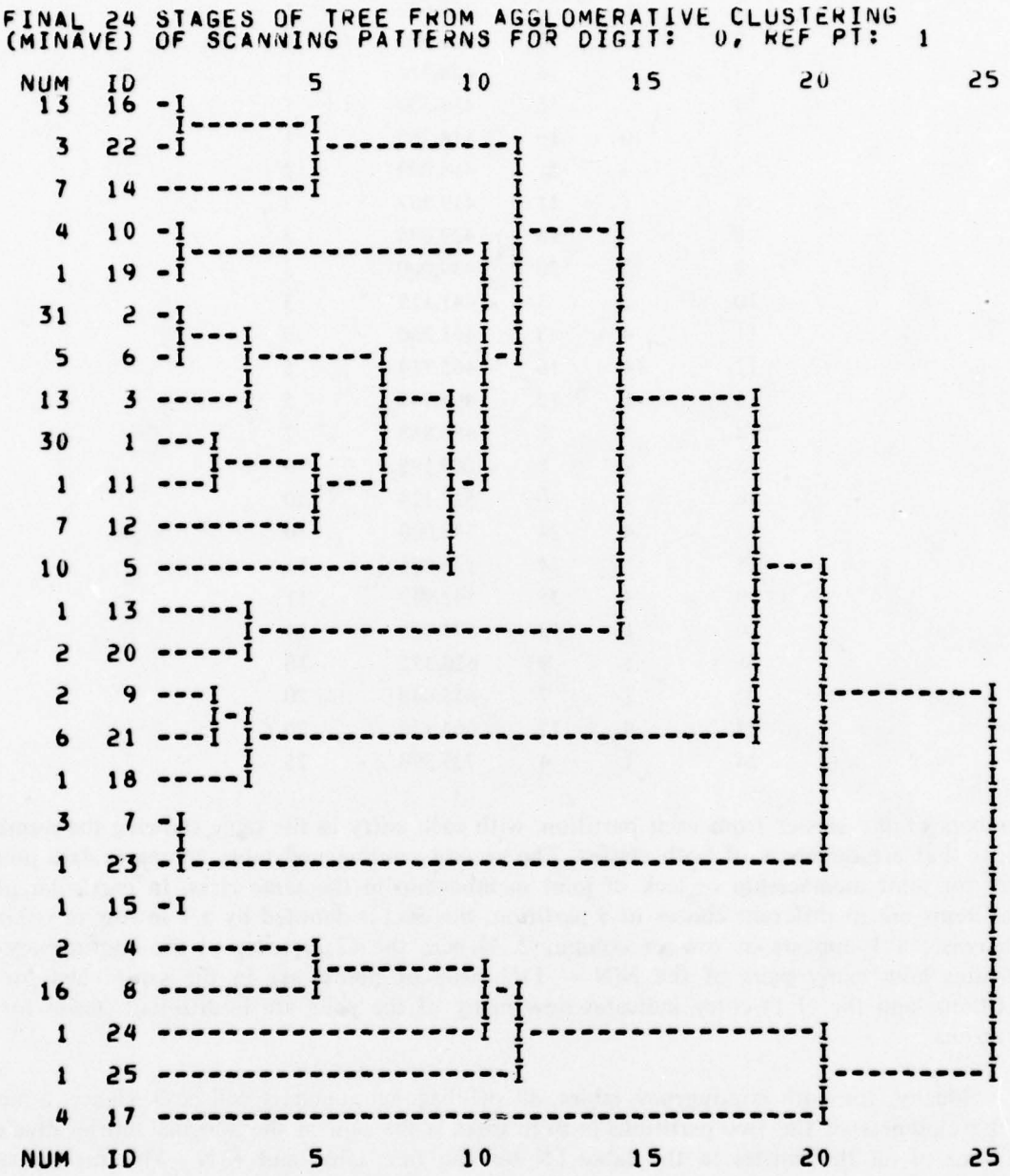


Figure 15. Tree for Final 24 Stages of MINAVE Agglomerative Clustering of All (166) Scanning Patterns for First Reference Point of Digit "Zero"

STATISTICS FOR DIGIT: 0; REF PT: 1; NO OF DATA PTS: 166
MINAVE AGGLOM CLUSTERING; 23 FEB79
PRE AND POST ITERATIVE OPTIMIZATION FOR MIN JE

C	ERR	NO OF ITERS	JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
			PRE	POST	PRE	POST	PRE	POST
1	0.0	0	46284.8	46284.9	0.120	0.125	0.000	0.000
2	729.4	8	40730.9	40504.1	0.034	0.118	0.136	0.143
3	664.1	80	39361.8	35711.4	0.032	0.057	0.176	0.296
4	655.0	93	38118.8	33671.6	0.050	0.061	0.214	0.375
5	626.3	139	36216.8	31628.2	0.016	0.036	0.278	0.463
6	573.3	124	35626.9	30504.0	0.011	0.046	0.299	0.517
7	542.9	141	35252.2	29106.6	0.066	0.033	0.313	0.590
8	538.1	96	32933.0	28154.5	0.011	0.027	0.405	0.644
9	530.0	80	32559.9	27385.4	0.024	0.018	0.422	0.690
10	527.1	75	31768.7	26902.9	0.040	*******	0.457	0.720
11	509.2	-1	30486.6	********	0.092	0.000	0.518	-0.000
12	488.9	-1	27668.6	********	0.041	0.000	0.673	-0.000
13	464.7	-1	26542.2	********	0.031	0.000	0.744	-0.000
14	462.7	-1	25724.7	********	0.017	0.000	0.799	-0.000
15	461.3	-1	25276.9	********	0.053	0.000	0.831	-0.000
16	441.1	-1	23934.5	********	0.010	0.000	0.934	-0.000
17	439.0	-1	23703.2	********	0.011	0.000	0.953	-0.000
18	429.9	-1	23454.0	********	0.011	0.000	0.973	-0.000
19	420.0	-1	23186.8	********	0.016	0.000	0.996	-0.000
20	418.0	-1	22808.2	********	0.010	0.000	1.029	-0.000
21	414.8	-1	22570.3	********	0.010	0.000	1.051	-0.000
22	414.5	-1	22343.0	********	0.027	0.000	1.072	-0.000
23	408.7	-1	21729.7	********	0.011	0.000	1.130	-0.000
24	406.0	-1	21497.5	********	0.018	0.000	1.155	-0.000
25	403.7	-1	21114.8	**********	*******	0.000	1.192	-0.000

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
			PRE	POST	PRE	POST	PRE	POST
1	0.000	0.000	0.000	0.000	19.799	20.608	0.000	0.000
2	0.135	0.141	0.579	0.499	5.512	19.406	22.363	25.406
3	0.130	0.218	1.042	0.614	5.147	9.310	14.334	24.131
4	0.139	0.244	1.114	0.616	8.083	9.831	11.568	20.228
5	0.169	0.281	1.153	0.644	2.622	5.723	11.189	18.652
6	0.173	0.299	1.061	0.666	1.683	7.330	9.575	16.555
7	0.175	0.330	1.409	0.702	10.460	5.201	8.293	15.640
8	0.221	0.350	1.327	0.772	1.790	4.316	9.151	14.535
9	0.224	0.367	1.670	0.797	3.815	2.766	8.273	13.544
10	0.239	0.376	1.585	0.864	6.296	*******	7.920	12.488
11	0.266	-0.000	1.533	********	14.327	0.000	8.032	-0.000
12	0.340	-0.000	1.479	********	6.770	0.000	9.420	-0.000
13	0.371	-0.000	1.505	********	4.712	0.000	9.484	-0.000
14	0.394	-0.000	1.517	********	2.646	0.000	9.345	-0.000
15	0.405	-0.000	1.504	********	8.019	0.000	8.964	-0.000
16	0.450	-0.000	1.493	********	1.450	0.000	9.338	-0.000
17	0.454	-0.000	1.664	********	1.566	0.000	8.872	-0.000
18	0.459	-0.000	1.886	********	1.687	0.000	8.475	-0.000
19	0.466	-0.000	2.063	********	2.400	0.000	8.135	-0.000
20	0.476	-0.000	2.127	********	1.523	0.000	7.909	-0.000
21	0.482	-0.000	2.301	********	1.460	0.000	7.618	-0.000
22	0.487	-0.000	2.458	********	3.953	0.000	7.348	-0.000
23	0.509	-0.000	2.392	********	1.528	0.000	7.345	-0.000
24	0.515	-0.000	2.603	********	2.528	0.000	7.119	-0.000
25	0.527	-0.000	2.554	********	*******	0.000	7.003	-0.000

Figure 16. Parameter Comparisons for Pre- and Postiterative Optimization of MINAVE Partitions Using All Points

STATISTICS FOR DIGIT: 0; REF PT: 1; NO OF DATA PTS: 158
MINAVE AGGLOM CLUSTERING; 23 FEB79
PRE AND POST ITERATIVE OPTIMIZATION FOR MIN JE

C	ERR	NO OF ITERS	JE (=TR(W)) PRE	JE (=TR(W)) POST	JE(C)-JE(C+1)/JE(C) PRE	JE(C)-JE(C+1)/JE(C) POST	TR(B)/TR(W) PRE	TR(B)/TR(W) POST
1	0.0	0	42925.1	46284.9	0.124	0.193	0.000	0.000
2	718.9	7	37601.3	37357.9	0.027	0.118	0.142	0.259
3	681.5	79	36595.8	32946.4	0.038	0.067	0.173	0.405
4	651.6	101	35214.9	30736.0	0.079	0.059	0.219	0.506
5	562.4	83	32448.7	28913.7	0.076	0.043	0.323	0.601
6	536.4	74	29969.3	27657.9	0.042	0.029	0.432	0.673
7	497.6	56	28697.2	26863.4	0.030	0.035	0.496	0.723
8	477.1	62	27835.6	25914.2	0.037	0.057	0.542	0.786
9	472.6	65	26819.1	24448.6	0.017	0.033	0.601	0.893
10	461.3	73	26371.3	23648.0	0.014	*******	0.628	0.957
11	451.4	-1	26008.9	********	0.085	0.000	0.650	-0.000
12	447.5	-1	23792.7	********	0.032	0.000	0.804	-0.000
13	437.0	-1	23037.6	********	0.024	0.000	0.863	-0.000
14	435.2	-1	22489.4	********	0.017	0.000	0.909	-0.000
15	418.0	-1	22110.8	********	0.028	0.000	0.941	-0.000
16	408.7	-1	21497.5	********	0.018	0.000	0.997	-0.000
17	403.7	-1	21114.8	********	-0.111	0.000	1.033	-0.000

C	C(N-C)TR(B)/2N(C-1)TR(W)		BTL'S SIGMA PRE	BTL'S SIGMA POST	(N-C)*DELJE/JE(C) PRE	(N-C)*DELJE/JE(C) POST	(N-C)*TR(B)/(C-1)*TR(W) PRE	(N-C)*TR(B)/(C-1)*TR(W) POST
1	0.000	0.000	0.000	0.000	19.472	30.281	0.000	0.000
2	0.133	0.225	0.630	0.552	4.172	18.422	22.087	37.277
3	0.121	0.284	1.040	0.664	5.849	10.399	13.404	31.376
4	0.135	0.313	1.371	0.638	12.097	9.130	11.239	25.969
5	0.186	0.346	1.323	0.675	11.690	6.645	12.349	22.980
6	0.238	0.370	1.204	0.733	6.452	4.366	13.142	20.474
7	0.263	0.384	1.130	0.942	4.534	5.336	12.477	18.195
8	0.280	0.406	1.130	0.976	5.478	8.483	11.616	16.845
9	0.303	0.451	1.150	0.921	2.488	4.879	11.185	16.635
10	0.311	0.474	1.173	1.113	2.034	*******	10.322	15.741
11	0.317	-0.000	1.203	********	12.526	0.000	9.561	-0.000
12	0.386	-0.000	1.176	********	4.634	0.000	10.673	-0.000
13	0.408	-0.000	1.201	********	3.450	0.000	10.451	-0.000
14	0.424	-0.000	1.186	********	2.424	0.000	10.065	-0.000
15	0.434	-0.000	1.285	********	3.967	0.000	9.615	-0.000
16	0.455	-0.000	1.280	********	2.528	0.000	9.436	-0.000
17	0.466	-0.000	1.296	********	-15.621	0.000	9.103	-0.000

Figure 17. Parameter Comparisons for Pre- and Postiterative Optimization of MINAVE Partitions With Outliers Discarded

STATISTICS FOR DIGIT: 0; REF PT: 1; NO OF DATA PTS: 166
MINMAX AGGLOM CLUSTERING; 23 FEB79
PRE AND POST ITERATIVE OPTIMIZATION FOR MIN JE

C	ERR	NO OF ITERS	JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
			PRE	POST	PRE	POST	PRE	POST
1	0.0	0	46284.9	46284.9	0.075	0.133	0.000	0.000
2	1421.0	54	42816.1	40134.8	0.139	0.110	0.081	0.153
3	1331.0	23	36874.7	35734.2	0.024	0.056	0.255	0.295
4	1091.0	76	35985.7	33736.4	0.049	0.060	0.286	0.372
5	1020.0	84	34208.1	31726.3	0.045	0.044	0.353	0.459
6	976.0	114	32673.8	30345.3	0.046	0.019	0.417	0.525
7	862.0	44	31180.5	29771.7	0.028	0.047	0.484	0.555
8	855.0	68	30303.8	28361.9	0.034	0.038	0.527	0.632
9	844.0	60	29284.5	27293.4	0.028	0.029	0.581	0.696
10	805.0	48	28472.8	26491.2	0.038*******		0.626	0.747
11	804.0	-1	27397.5********		0.027	0.000	0.689	-0.000
12	785.0	-1	26647.0********		0.041	0.000	0.737	-0.000
13	756.0	-1	25557.7********		0.023	0.000	0.811	-0.000
14	712.0	-1	24964.9********		0.019	0.000	0.854	-0.000
15	684.0	-1	24492.6********		0.035	0.000	0.890	-0.000
16	682.0	-1	23628.7********		0.024	0.000	0.959	-0.000
17	678.0	-1	23068.7********		0.040	0.000	1.006	-0.000
18	678.0	-1	22152.5********		0.027	0.000	1.089	-0.000
19	606.0	-1	21544.2********		0.014	0.000	1.148	-0.000
20	597.0	-1	21245.6********		0.016	0.000	1.179	-0.000
21	593.0	-1	20908.4********		0.017	0.000	1.214	-0.000
22	584.0	-1	20545.1********		0.029	0.000	1.253	-0.000
23	571.0	-1	19946.5********		0.020	0.000	1.320	-0.000
24	553.0	-1	19543.8********		0.028	0.000	1.368	-0.000
25	537.0	-1	19000.4**********		**********	0.000	1.436	-0.000

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
			PRE	POST	PRE	POST	PRE	POST
1	0.000	0.000	0.000	0.000	12.366	21.924	0.000	0.000
2	0.080	0.151	0.167	0.305	22.757	17.982	13.286	25.131
3	0.188	0.217	0.545	0.627	3.930	9.113	20.798	24.063
4	0.186	0.242	0.685	0.633	8.002	9.652	15.455	20.086
5	0.214	0.278	0.696	0.622	7.221	7.008	14.210	18.470
6	0.241	0.304	0.701	0.663	7.313	3.024	13.330	16.809
7	0.271	0.310	0.696	0.816	4.471	7.530	12.837	14.698
8	0.287	0.344	0.683	0.839	5.314	5.952	11.903	14.264
9	0.309	0.370	0.764	0.880	4.352	4.614	11.393	13.656
10	0.327	0.390	0.794	0.891	5.891*******		10.843	12.951
11	0.354	-0.000	0.999********		4.246	0.000	10.685	-0.000
12	0.373	-0.000	1.067********		6.296	0.000	10.317	-0.000
13	0.405	-0.000	1.088********		3.549	0.000	10.340	-0.000
14	0.421	-0.000	1.108********		2.876	0.000	9.985	-0.000
15	0.434	-0.000	1.138********		5.326	0.000	9.597	-0.000
16	0.462	-0.000	1.187********		3.555	0.000	9.588	-0.000
17	0.480	-0.000	1.169********		5.918	0.000	9.372	-0.000
18	0.514	-0.000	1.215********		4.064	0.000	9.484	-0.000
19	0.537	-0.000	1.223********		2.038	0.000	9.378	-0.000
20	0.546	-0.000	1.253********		2.317	0.000	9.056	-0.000
21	0.557	-0.000	1.428********		2.519	0.000	8.799	-0.000
22	0.569	-0.000	1.610********		4.195	0.000	8.591	-0.000
23	0.595	-0.000	1.594********		2.887	0.000	8.583	-0.000
24	0.611	-0.000	1.628********		3.948	0.000	8.448	-0.000
25	0.635	-0.000	1.616********		*******	0.000	8.436	-0.000

Figure 18. Parameter Comparisons for Pre- and Postiterative Optimization of MINMAX Partitions Using All Points

STATISTICS FOR DIGIT: 0; REF PT: 1; NO OF DATA PTS: 164
MINMAX AGGLOM CLUSTERING; 23 FEB79
PRE AND POST ITERATIVE OPTIMIZATION FOR MIN JE

C	ERR	NO OF ITERS	JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
			PRE	POST	PRE	POST	PRE	POST
1	0.0	0	45290.7	46284.9	0.077	0.152	0.000	0.000
2	1421.0	53	41788.0	39270.7	0.135	0.109	0.084	0.179
3	1331.0	24	36134.4	34983.8	0.049	0.066	0.253	0.323
4	1020.0	49	34356.9	32676.7	0.045	0.052	0.318	0.416
5	976.0	53	32822.5	30969.4	0.045	0.041	0.380	0.495
6	862.0	43	31329.2	29693.0	0.044	0.043	0.446	0.559
7	857.0	49	29961.7	28428.1	0.029	0.043	0.512	0.628
8	855.0	75	29085.0	27213.1	0.035	0.035	0.557	0.701
9	844.0	60	28065.7	26254.5	0.029	0.031	0.614	0.763
10	805.0	44	27254.0	25428.2	0.028	*******	0.662	0.820
11	785.0	-1	26503.5	********	0.041	0.000	0.709	-0.000
12	756.0	-1	25414.2	********	0.023	0.000	0.782	-0.000
13	712.0	-1	24821.4	********	0.019	0.000	0.825	-0.000
14	684.0	-1	24349.1	********	0.035	0.000	0.860	-0.000
15	682.0	-1	23485.2	********	0.024	0.000	0.928	-0.000
16	678.0	-1	22925.2	********	0.040	0.000	0.976	-0.000
17	678.0	-1	22009.0	********	0.025	0.000	1.058	-0.000
18	643.0	-1	21452.0	********	0.028	0.000	1.111	-0.000
19	606.0	-1	20843.7	********	0.014	0.000	1.173	-0.000
20	597.0	-1	20545.1	********	0.029	0.000	1.204	-0.000
21	571.0	-1	19946.5	********	0.020	0.000	1.271	-0.000
22	553.0	-1	19543.8	********	0.028	0.000	1.317	-0.000
23	537.0	-1	19000.4	********	-0.029	0.000	1.384	-0.000

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
			PRE	POST	PRE	POST	PRE	POST
1	0.000	0.000	0.000	0.000	12.606	24.702	0.000	0.000
2	0.082	0.174	0.172	0.305	21.917	17.684	13.579	28.935
3	0.184	0.235	0.573	0.645	7.920	10.618	20.398	26.005
4	0.204	0.268	0.567	0.620	7.145	8.359	16.973	22.211
5	0.227	0.296	0.572	0.649	7.234	6.553	15.100	19.658
6	0.254	0.319	0.574	0.676	6.897	6.731	14.082	17.657
7	0.282	0.347	0.866	0.742	4.594	6.710	13.387	16.436
8	0.299	0.376	0.820	0.832	5.467	5.495	12.417	15.619
9	0.322	0.401	0.899	0.846	4.483	4.878	11.891	14.782
10	0.341	0.423	0.911	0.897	4.241	*******	11.324	14.035
11	0.359	-0.000	0.985	********	6.289	0.000	10.845	-0.000
12	0.391	-0.000	1.007	********	3.545	0.000	10.807	-0.000
13	0.406	-0.000	1.028	********	2.873	0.000	10.377	-0.000
14	0.418	-0.000	1.068	********	5.322	0.000	9.924	-0.000
15	0.446	-0.000	1.118	********	3.553	0.000	9.882	-0.000
16	0.464	-0.000	1.105	********	5.915	0.000	9.626	-0.000
17	0.498	-0.000	1.152	********	3.720	0.000	9.719	-0.000
18	0.517	-0.000	1.257	********	4.140	0.000	9.544	-0.000
19	0.541	-0.000	1.263	********	2.077	0.000	9.448	-0.000
20	0.550	-0.000	1.293	********	4.195	0.000	9.128	-0.000
21	0.575	-0.000	1.287	********	2.887	0.000	9.085	-0.000
22	0.590	-0.000	1.329	********	3.948	0.000	8.908	-0.000
23	0.614	-0.000	1.326	********	-4.033	0.000	8.868	-0.000

Figure 19. Parameter Comparisons for Pre- and Postiterative Optimization of MINMAX Partitions With Outliers Discarded

STATISTICS FOR DIGIT: 0; REF PT: 1; NO OF DATA PTS: 166
MINAVE AND MINMAX AGGLOM CLUSTERING; 23 FEB79
PRE ITERATIVE OPTIMIZATION FOR MIN JE

C	ERR		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.0	0.0	46284.8	46284.9	0.120	0.075	0.000	0.000
2	729.4	1421.0	40730.9	42816.1	0.034	0.139	0.136	0.081
3	664.1	1331.0	39361.8	36874.7	0.032	0.024	0.176	0.255
4	655.0	1091.0	38118.8	35985.7	0.050	0.049	0.214	0.286
5	626.3	1020.0	36216.8	34208.1	0.016	0.045	0.278	0.353
6	573.3	976.0	35626.9	32673.8	0.011	0.046	0.299	0.417
7	542.9	862.0	35252.2	31180.5	0.066	0.028	0.313	0.484
8	538.1	855.0	32933.0	30303.8	0.011	0.034	0.405	0.527
9	530.0	844.0	32559.9	29284.5	0.024	0.028	0.422	0.581
10	527.1	805.0	31768.7	28472.8	0.040	0.038	0.457	0.626
11	509.2	804.0	30486.6	27397.5	0.092	0.027	0.518	0.689
12	488.9	785.0	27668.6	26647.0	0.041	0.041	0.673	0.737
13	464.7	756.0	26542.2	25557.7	0.031	0.023	0.744	0.811
14	462.7	712.0	25724.7	24964.9	0.017	0.019	0.799	0.854
15	461.3	684.0	25276.9	24492.6	0.053	0.035	0.831	0.890
16	441.1	682.0	23934.5	23628.7	0.010	0.024	0.934	0.959
17	439.0	678.0	23703.2	23068.7	0.011	0.040	0.953	1.006
18	429.9	678.0	23454.0	22152.5	0.011	0.027	0.973	1.089
19	420.0	606.0	23186.8	21544.2	0.016	0.014	0.996	1.148
20	418.0	597.0	22808.2	21245.6	0.010	0.016	1.029	1.179
21	414.8	593.0	22570.3	20908.4	0.010	0.017	1.051	1.214
22	414.5	584.0	22343.0	20545.1	0.027	0.029	1.072	1.253
23	408.7	571.0	21729.7	19946.5	0.011	0.020	1.130	1.320
24	406.0	553.0	21497.5	19543.8	0.018	0.028	1.153	1.368
25	403.7	537.0	21114.8	19000.4	**********	**********	1.192	1.436

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	19.559	12.216	0.000	0.000
2	0.133	0.079	0.579	0.167	5.445	22.480	22.090	13.124
3	0.128	0.186	1.042	0.545	5.084	3.882	14.158	20.543
4	0.138	0.184	1.114	0.685	7.984	7.903	11.425	15.264
5	0.166	0.211	1.153	0.696	2.590	7.132	11.050	14.033
6	0.171	0.238	1.061	0.701	1.662	7.221	9.453	13.164
7	0.173	0.267	1.409	0.696	10.329	4.414	8.189	12.676
8	0.218	0.283	1.327	0.683	1.768	5.247	9.035	11.753
9	0.221	0.305	1.670	0.764	3.766	4.296	8.167	11.248
10	0.235	0.322	1.585	0.794	6.215	5.816	7.819	10.704
11	0.263	0.349	1.533	0.999	14.142	4.191	7.929	10.548
12	0.336	0.368	1.479	1.067	6.188	6.214	9.297	10.183
13	0.366	0.400	1.505	1.088	4.651	3.502	9.360	10.205
14	0.389	0.416	1.517	1.108	2.611	2.838	9.222	9.854
15	0.400	0.428	1.504	1.138	7.913	5.256	8.845	9.469
16	0.444	0.456	1.493	1.187	1.430	3.507	9.214	9.461
17	0.448	0.473	1.664	1.169	1.545	5.838	8.753	9.246
18	0.453	0.507	1.886	1.215	1.664	4.009	8.360	9.356
19	0.459	0.529	2.063	1.223	2.367	2.010	8.025	9.251
20	0.470	0.538	2.127	1.253	1.502	2.265	7.801	8.932
21	0.475	0.549	2.301	1.428	1.440	2.485	7.512	8.678
22	0.480	0.561	2.458	1.610	3.898	4.137	7.246	8.472
23	0.502	0.586	2.392	1.594	1.506	2.847	7.242	8.463
24	0.507	0.602	2.603	1.628	2.492	3.893	7.018	8.329
25	0.520	0.626	2.554	1.616	**********	**********	6.904	8.317

Figure 20. Parameter Comparisons for Preiterative Optimization of MINAVE and MINMAX Partitions Using All Points

STATISTICS FOR DIGIT: 0; REF PT: 1; NO OF DATA PTS: 100
MINAVE AND MINMAX AGGLOM CLUSTERING; 23 FEB79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	46284.9	46284.9	0.125	0.133	0.000	0.000
2	8	54	40504.1	40134.8	0.118	0.110	0.143	0.153
3	80	23	35711.4	35734.2	0.057	0.056	0.296	0.295
4	93	76	33671.6	33736.4	0.061	0.060	0.375	0.372
5	139	84	31628.2	31726.3	0.036	0.044	0.463	0.459
6	124	114	30504.0	30345.3	0.046	0.019	0.517	0.525
7	141	44	29106.6	29771.7	0.033	0.047	0.590	0.555
8	96	68	28154.5	28361.9	0.027	0.038	0.644	0.632
9	80	60	27385.4	27293.4	0.018	0.029	0.690	0.696
10	75	48	26902.9	26491.2	**********		0.720	0.747

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	20.358	21.659	0.000	0.000
2	0.139	0.150	0.499	0.305	19.169	17.763	23.121	24.824
3	0.215	0.215	0.614	0.627	9.196	9.001	23.835	23.768
4	0.241	0.239	0.616	0.633	9.710	9.533	19.979	19.858
5	0.277	0.275	0.644	0.622	5.652	6.921	18.420	18.241
6	0.295	0.300	0.666	0.663	7.238	2.986	16.348	16.599
7	0.326	0.306	0.702	0.816	5.135	7.435	15.443	14.514
8	0.346	0.339	0.772	0.839	4.262	5.877	14.351	14.083
9	0.362	0.365	0.797	0.880	2.731	4.556	13.371	13.482
10	0.371	0.385	0.864	0.891	**********		12.328	12.785

Figure 21. Parameter Comparisons for Postiterative Optimization of MINAVE and MINMAX Partitions Using All Points

After all class assignments and contingency tables have been printed, the measure just described is printed in a summary listing for the partition comparisons both before and after optimization (Figure 23).

E. OBSERVATIONS ON THE CLUSTERING RESULTS

Since the conclusions to be presented in this subsection have not been proved analytically but have only been observed from the clustering results, they are presented as observations only. However, these observations are made on the basis of the results of clustering 34 different pattern types, which would imply some generality to the observations.

In investigating the properties of various criteria, it is first useful to have a measure of the distribution of the class size, i.e., whether most of the samples are in one class or whether they are evenly distributed in all classes. The measure used was a normalized version of the entropy given by

$$E = -(1/EMAX) \sum_{i=1}^{c} (n_i/n) \log_2 (n_i/n) \qquad (19)$$

where

$$n = \sum_{i=1}^{c} n_i$$

```
                MINAVE CLASSES
    1    2   15   16   17   22   25   26   27   34   35   38   39   41
        47   54   55   74   75   90   91  100  101  102  103  106  126
       127  128  133  144  147  149  150  159  165  166
    2   13   19   21   32   48   49   56   64   72   85   88   98   99
       105  108  110  111  121  148
    3    1   11   12   18   24   36   37   45   61   96   97  114  115
       129  130  131  136  141  145
    4   14   52   89  112  113  138  139  155  161
    5    9   20   23   28   43   59   70   73   76   77   79   80   81
        84  124  125  142  156  158  164
    6   10   31   78   87  107  117  137
    7   33   50   51   53   58   62   63   66   67   71   82   85   86
        93   94  118  119  151  162  163
    8   30   44   46   60  122  132  134  135  140
    9    3    4    5    6    7    8   29   40   42   57   65   68   69
        92   95  104  109  116  143  146  152  153  154  157  160
   10  120  123
**********

                MINMAX CLASSES
    1    1   15   16   17   18   22   24   25   34   35   41   45   47
        54   55   61   74   96  100  101  106  126  128  129  130  131
       133  144  145  165  166
    2    2   26   27   38   39   75   90   91  102  103  127  146  147
       149  150  159
    3    3    4    5    6    7    8   29   40   42   57   65   68   69
        92   95  104  109  116  143  152  153  154  157  160
    4   11   12   30   36   37   44   46   60   97  114  115  122  132
       134  135  136  140  141
    5    9   20   23   28   43   59   70   73   76   77   79   80   81
        84  124  125  142  156  158  164
    6   10   31   78   87  107  117  137
    7   13   19   21   32   49   56   64   85   99  108  110  111
    8   33   50   51   53   58   62   63   66   67   71   82   83   93
        94  118  119  151  162  163
    9   48   72   88   98  105  121  123  148
   10   14   52   86   89  112  113  120  138  139  155  161
**********

C1:             MINMAX CLASSES

                  3    1    5    8    7   10    4    6    9    2
                 --   --   --   --   --   --   --   --   --   --
MINAVE    9     24    0    0    0    0    0    0    0    0    1    0
CLASSES   1      0   21    0    0    0    0    0    0    0   15    0
          5      0    0   20    0    0    0    0    0    0    0    0
          7      0    0    0   19    0    1    0    0    0    0    0
          2      0    0    0    0   12    0    0    0    7    0    0
          4      0    0    0    0    0    9    0    0    0    0    0
          8      0    0    0    0    0    0    9    0    0    0    0
          6      0    0    0    0    0    0    0    7    0    0    0
         10      0    0    0    0    0    1    0    0    1    0    0
          3      0   10    0    0    0    0    9    0    0    0    0
                 0    0    0    0    0    0    0    0    0    0    0

C2:             MINMAX CLASSES

                      1       2
                   ------  ------
MINAVE     1      11618     332
CLASSES    2        533    1213
```

Figure 22. Class Assignments and Contingency Tables for Postiterative *Optimization of* 10 Classes Using All Points for Reference Point 1 of Digit 0

PRE ITERATIVE OPTIMIZATION FOR MIN JE

NC:	2	3	4	5	6	7	8	9	10
C1:	0.440	0.548	0.542	0.536	0.470	0.470	0.392	0.416	0.434
C2:	0.504	0.620	0.618	0.598	0.573	0.540	0.643	0.651	0.676

POST ITERATIVE OPTIMIZATION FOR MIN JE

NC:	2	3	4	5	6	7	8	9	10
C1:	0.633	0.964	0.657	0.795	0.699	0.627	0.633	0.651	0.735
C2:	0.532	0.943	0.736	0.857	0.869	0.870	0.887	0.899	0.937

Figure 23. Contingency Table Measures for Pre- and Postiterative Optimization Using All Points for Reference Point 1 of Digit 0

and EMAX = $\log_2 c$. This function is maximum for c equally divided clusters having $n_i = n/c$, and is minimum for $(c - 1)$ $n_i's = 1$ clusters having one cluster $n_i = (n - c - 1)/n$.

Observation 1: The first observation is experimental confirmation of the known fact in cluster analysis that the minimization of J_e, the sum-of-squared error, favors equal sized clusters, an example of which is given in Figure B-1. One demonstration of why this is true is given in Subsection IV.B. To demonstrate in another way, remember that minimizing J_e is equivalent to maximizing tr S_B, given by

$$\text{tr } S_B = \frac{1}{2n} \sum_{i=1}^{c} \sum_{j=1}^{c} n_i n_j \|\vec{m}_i - \vec{m}_j\|^2 \tag{20}$$

Consider just one of the terms. Assuming that the means remain relatively constant as points are changed between class i and class j, it is easy to show that $n_i n_j$ is maximum for $n_i = n_j$.

The results of the iterative optimization based on minimizing J_e in fact showed that E (the normalized entropy) increased after the optimization as shown by the histograms of E for the MINAVE agglomerative clustering in Figure 24 for the 24 scanning patterns (c = 2 through 10) and in Figure 25 for the 10 recognition patterns (c = 2 through 10).

Observation 2: MINMAX agglomerative clustering favors equal-sized clusters more than MINAVE agglomerative clustering. No previous reference to this type of observation could be found in the clustering literature, although many references existed to the "chaining" that occurs in agglomerative clustering when combining the two clusters having the minimum minimum distance between them.

Figure 26 shows the results as histograms of E for both the MINAVE and the MINMAX agglomerative clustering (before iterative optimization) for the 24 scanning patterns (c = 2 through 10). The results shown in Figure 27 are the same for the recognition patterns.

Observation 1 indicated that lower J_es (and, hence, larger αs) result from more equal-sized clusters, so that two "corollaries" exist for observation 2. The first is that J_e for MINMAX agglomerative clustering (preiterative optimization) is generally smaller than J_e for MINAVE agglomerative clustering. The second corollary is that, since J_e is usually lower for the MINMAX agglomeration, the number of iterations is also lower for the iterative optimization of the MINMAX agglomerative clusters. Histograms are not given, but refer to the columns for the number of iterations given in the 10 tables of postiterative optimization statistics for the 10 recognition patterns in Appendix D.

Observation 3: In spite of quite different starting partitions, the iterative optimization procedure applied to the MINAVE agglomerative clusters and to the MINMAX agglomerative clusters yielded similar partitions. A measure of similarity given by the ratio of the sum of the diagonal entries of a contingency table divided by the sum of all the entries in the table is shown in Table 7 for both scanning and recognition patterns before iterative optimization and in Table 8 for both pattern types after iterative optimization. These similarity measures are for the first type of contingency table described in Subsection IV.D.3.

Observation 4: Although α was larger after iterative optimization than before (as it should be since the optimization criterion is to minimize J_e, which is equivalent to maximizing α), σ almost always decreased for the optimization using the MINAVE agglomerative beginning partitions and decreased for about half the pattern types for optimization using the MINMAX agglomerative beginning partitions. Plots of $\sigma_{post}/\sigma_{pre}$ versus $\alpha_{post}/\alpha_{pre}$ are given in Figure 28 for the MINAVE clustering and in Figure 29 for the MINMAX clustering for scanning patterns.

Two important points should be made. The first point is that the σ decrease was much greater for the MINAVE case than for the MINMAX case because σ_{pre} was much larger for the former, as can be seen by comparing Figure 30 and Figure 31, which show σ_{pre} versus α_{pre} for scanning pattterns in both cases. As a matter of interest, Figure 32 shows σ versus α for postoptimization of the MINAVE agglomerative clustering partitions, showing that as the clusters approach equal sizes, as happens for the optimization (observation 1), σ and α approach the same value. In fact, for equally divided cluster partitions, $\sigma = \alpha$, as can be seen from the final two expressions in Subsection IV.C by setting $n_i = n_j = n/c$. The second point to remember is to temper the conclusions reached about the relationship between σ and α by the fact that optimization is with respect to α (actually J_e) while σ is being monitored only. Possibly different conclusions would be reached if iterative optimization were with respect to σ with α being monitored only.

F. TESTING CLUSTER VALIDITY WITH *A PRIORI* INFORMATION ABOUT DATA

The problem of testing cluster validity is a subject that has received very little attention in the literature, probably because of the difficulty of the problem. One of the few references is in Duda and Hart,[28] who use a hypothesis testing approach to test validity on the basis of the size of the reduction in J_e. In the specific example given, they assume multivariate normal distributions and advance the hypothesis that the data are actually from one cluster. They then derive an expression for testing this hypothesis for J_e to a specified significance level.

The approach taken in this section is an entirely different method for testing cluster validity. In the total voice speaker verification final report,[1] descriptions of the characteristics of the reference patterns generated from the clustering algorithm are given in terms of *a priori*

Figure 24. Histograms of Normalized Entropy as Measure of Dispersion in Class Sizes for MINAVE Agglomerative Clustering of Scanning Patterns

information known about the data points making up each class. In this study, a quantitative measure is used for the same type of comparison. Specifically, it is assumed that a male/female division of the data is a correct way to separate the data, and then the degree to which the actual clusters agree with this assumption is measured. Because of the differences in vocal tract resonances (formants) between males and females, this is a good assumption in most cases (probably a better assumption than assuming a unimodal distribution for the data). Reference to the total voice final report, however, reveals cases where the data actually cluster on the basis of other attributes such as the scanning patterns for the third reference point of "two" which, since the formants for /u/ for males and females are very close, splits according to context. This would suggest extending the technique in this subsection to account quantitatively for multiple attributes. This discussion, however, will consider only the male/female distinction.

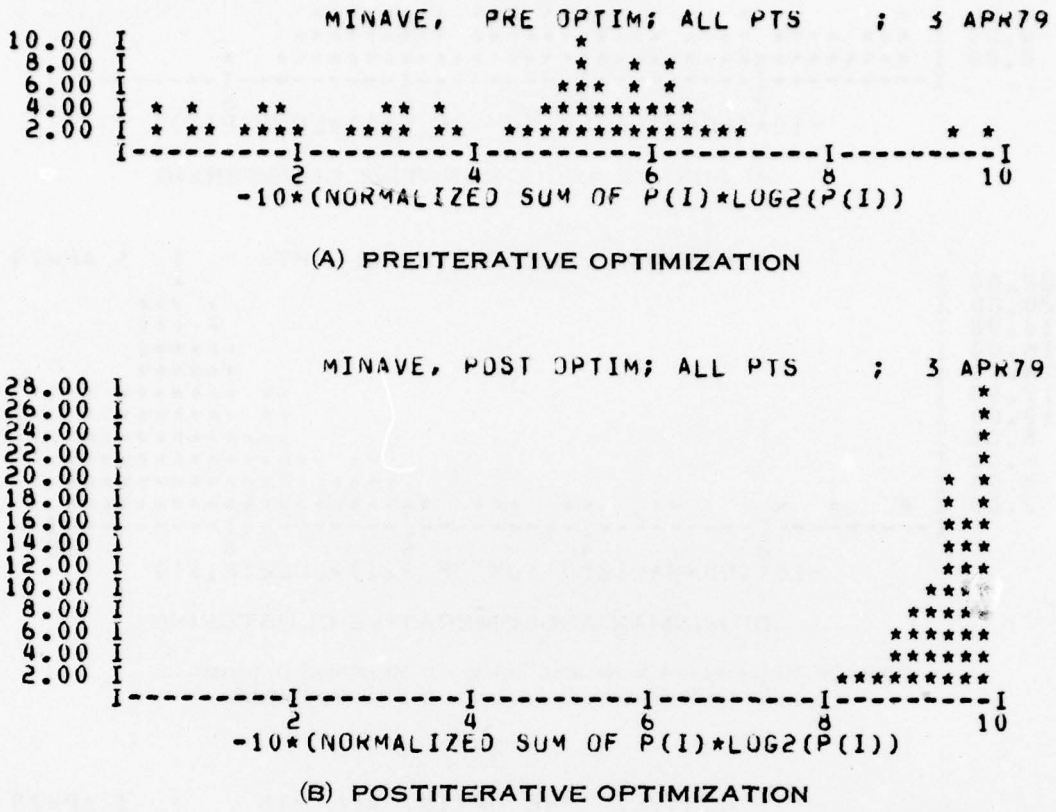


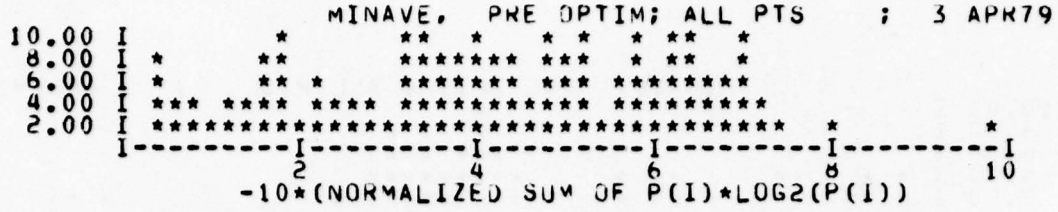
Figure 25. *Histograms of Normalized Entropy* as Measure of Dispersion in Class Sizes for MINAVE Agglomerative Clustering of Recognition Patterns

The proposal is that the average information gained by knowing in which class a point falls should be reduced by the *a priori* knowledge of an attribute of that point, if the classes represent that attribute. Alternately stated, the proposal is that the average uncertainty about the class in which a point falls should be reduced by the amount of certainty gained about the class membership, knowing the attribute (the sex in this case). From the information theory literature (e.g., Reza[33]), the average uncertainty is the entropy, given by*

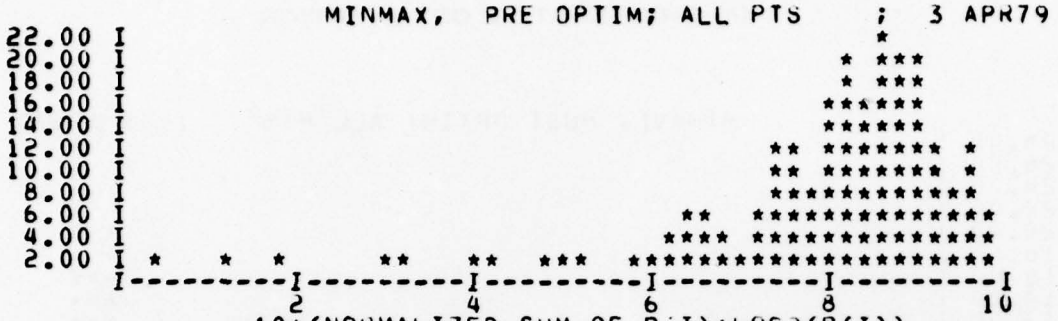
$$H(c) = -\sum_{i=1}^{c} p(i) \log p(i) \qquad (21)$$

[33] F.M. Reza, *An Introduction to Information Theory.* New York: McGraw-Hill, 1961.

*All logarithms are taken to base 2.

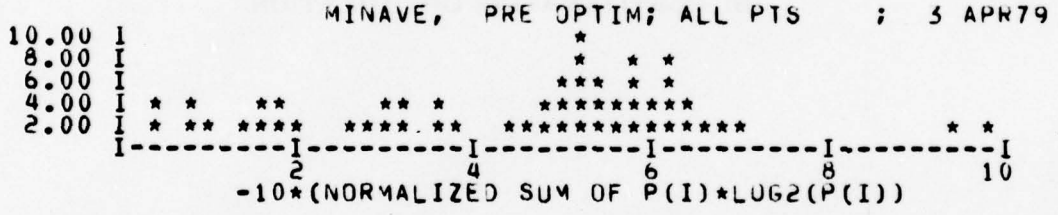


(A) MINAVE AGGLOMERATIVE CLUSTERING

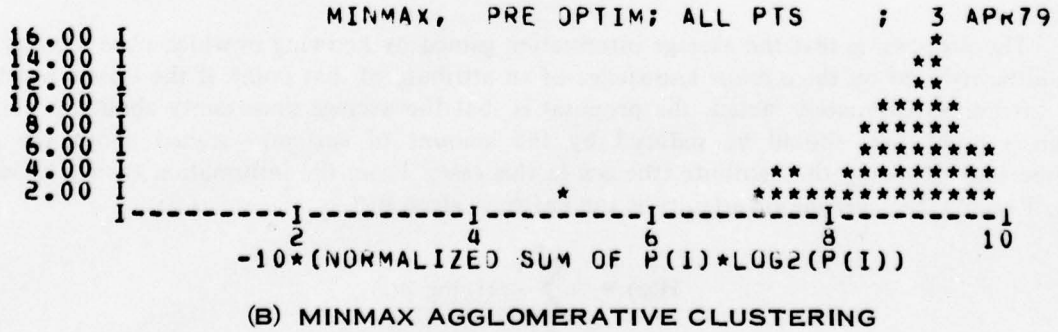


(B) MINMAX AGGLOMERATIVE CLUSTERING

Figure 26. Histograms of Normalized Entropy as Measure of Dispersion in Class Size for Preiterative Optimization of Scanning Patterns



(A) MINAVE AGGLOMERATIVE CLUSTERING



(B) MINMAX AGGLOMERATIVE CLUSTERING

Figure 27. Histograms of Normalized Entropy as Measure of Dispersion in Class Sizes for Preiterative Optimization of Recognition Patterns

TABLE 7. CONTINGENCY TABLE OF THE FIRST KIND*
RESULTS FOR PREITERATIVE OPTIMIZATION

For Scanning Patterns

Digit	Ref Point	Contingency Table Ratio for Given Number of Classes								
		2	3	4	5	6	7	8	9	10
0	1	0.440	0.548	0.542	0.536	0.470	0.470	0.392	0.416	0.434
0	2	0.663	0.536	0.548	0.373	0.518	0.536	0.524	0.542	0.524
0	3	1.000	0.928	0.934	0.825	0.536	0.500	0.488	0.506	0.416
1	1	0.815	0.643	0.655	0.756	0.750	0.720	0.744	0.643	0.643
1	2	0.964	0.833	0.661	0.661	0.661	0.524	0.554	0.357	0.369
2	1	0.929	0.911	0.702	0.708	0.452	0.470	0.554	0.399	0.399
2	2	0.765	0.756	0.524	0.286	0.500	0.482	0.482	0.583	0.476
2	3	0.613	0.821	0.821	0.714	0.744	0.738	0.732	0.804	0.798
3	1	0.858	0.657	0.639	0.509	0.633	0.609	0.533	0.568	0.586
3	2	0.592	0.432	0.438	0.456	0.456	0.604	0.639	0.544	0.533
4	1	0.916	0.940	0.725	0.689	0.533	0.533	0.401	0.467	0.467
4	2	1.000	0.641	0.641	0.641	0.461	0.461	0.551	0.617	0.605
5	1	0.615	0.538	0.544	0.527	0.509	0.669	0.657	0.663	0.675
5	2	0.609	0.396	0.391	0.391	0.556	0.604	0.556	0.651	0.639
6	1	0.970	0.707	0.719	0.551	0.527	0.587	0.611	0.557	0.563
6	2	0.569	0.575	0.389	0.587	0.593	0.581	0.479	0.473	0.533
6	3	0.665	0.563	0.671	0.599	0.539	0.581	0.539	0.551	0.599
7	1	0.994	0.756	0.583	0.417	0.429	0.429	0.417	0.393	0.399
7	2	0.881	0.821	0.607	0.500	0.518	0.435	0.536	0.548	0.548
7	3	0.827	0.536	0.524	0.571	0.494	0.494	0.589	0.565	0.530
8	1	0.798	0.512	0.464	0.494	0.589	0.423	0.393	0.446	0.440
8	2	0.583	0.679	0.571	0.601	0.565	0.589	0.417	0.417	0.458
9	1	0.515	0.544	0.562	0.373	0.527	0.462	0.485	0.497	0.544
9	2	0.621	0.538	0.538	0.580	0.592	0.621	0.627	0.633	0.716

For Recognition Patterns

Digit	Contingency Table Ratio for Given Number of Classes								
	2	3	4	5	6	7	8	9	10
0	0.807	0.578	0.590	0.434	0.434	0.416	0.440	0.434	0.349
1	0.750	0.607	0.619	0.530	0.435	0.363	0.387	0.375	0.369
2	0.500	0.500	0.482	0.494	0.542	0.536	0.512	0.524	0.411
3	0.633	0.556	0.408	0.604	0.544	0.491	0.509	0.456	0.479
4	0.772	0.790	0.790	0.515	0.491	0.491	0.497	0.491	0.497
5	0.858	0.775	0.763	0.728	0.604	0.633	0.633	0.491	0.491
6	0.778	0.707	0.605	0.617	0.593	0.587	0.599	0.587	0.689
7	0.494	0.821	0.750	0.565	0.583	0.583	0.595	0.560	0.583
8	0.554	0.464	0.631	0.619	0.518	0.524	0.548	0.464	0.548
9	0.491	0.538	0.556	0.574	0.396	0.361	0.320	0.408	0.432

*Refer to Subsection IV.D.3.

TABLE 8. CONTINGENCY TABLE OF THE FIRST KIND*
RESULTS FOR POSTITERATIVE OPTIMIZATION

For Scanning Patterns

Digit	Ref Point	Contingency Table Ratio for Given Number of Classes								
		2	3	4	5	6	7	8	9	10
0	1	0.633	0.964	0.657	0.795	0.699	0.627	0.633	0.651	0.735
0	2	1.000	1.000	0.723	0.807	0.590	0.596	0.596	0.741	0.753
0	3	1.000	0.669	1.000	0.614	0.687	0.699	0.633	0.669	0.657
1	1	1.000	0.857	1.000	0.863	0.631	0.458	0.518	0.732	0.827
1	2	0.958	0.815	0.554	0.702	0.548	0.708	0.655	0.708	0.702
2	1	0.917	0.815	0.625	0.708	0.708	0.685	0.655	0.571	0.571
2	2	1.000	1.000	0.637	0.726	0.857	0.810	0.738	0.732	0.714
2	3	1.000	0.696	1.000	0.571	0.756	0.917	0.869	0.935	0.899
3	1	0.675	0.775	0.462	0.799	0.917	0.663	0.568	0.663	0.669
3	2	0.988	0.633	0.970	0.716	0.686	0.716	0.734	0.710	0.704
4	1	1.000	1.000	0.856	0.826	0.587	0.689	0.695	0.766	0.641
4	2	1.000	0.808	0.665	0.665	0.605	0.641	0.719	0.743	0.635
5	1	0.675	0.503	0.923	0.799	0.893	0.716	0.462	0.793	0.586
5	2	1.000	0.964	0.959	0.846	0.710	0.941	0.822	0.675	0.675
6	1	1.000	1.000	0.844	0.707	0.880	0.850	0.665	0.599	0.635
6	2	1.000	0.647	0.850	0.737	0.814	0.665	0.575	0.713	0.713
6	3	1.000	0.467	0.689	1.000	0.886	0.629	0.701	1.000	0.904
7	1	1.000	0.988	0.500	0.929	0.750	0.601	0.720	0.655	0.726
7	2	1.000	0.577	0.762	0.875	0.964	0.821	0.702	0.607	0.667
7	3	0.488	0.988	0.982	0.601	0.613	0.714	0.661	0.798	0.756
8	1	1.000	0.839	0.940	0.685	0.488	0.470	0.607	0.690	0.661
8	2	0.952	0.685	0.637	0.673	0.679	0.571	0.851	0.714	0.690
9	1	1.000	0.781	0.680	0.645	0.787	0.775	0.686	0.746	0.917
9	2	0.692	0.757	0.846	1.000	0.746	0.716	0.799	0.805	0.870

For Recognition Patterns

Digit	Contingency Table Ratio for Given Number of Classes								
	2	3	4	5	6	7	8	9	10
0	1.000	0.753	0.789	0.590	0.639	0.861	0.783	0.753	0.614
1	0.940	0.738	0.940	0.780	0.655	0.661	0.792	0.827	0.685
2	1.000	0.964	0.798	0.851	0.708	0.744	0.696	0.565	0.607
3	1.000	0.645	0.686	0.562	0.556	0.515	0.562	0.562	0.592
4	0.988	0.916	0.491	0.617	0.563	0.743	0.587	0.563	0.647
5	1.000	0.834	0.728	0.817	0.609	0.580	0.734	0.757	0.852
6	1.000	0.982	0.743	0.796	0.832	0.689	0.796	0.629	0.593
7	1.000	0.929	0.726	0.530	0.679	0.554	0.714	0.655	0.601
8	1.000	1.000	1.000	0.958	0.839	0.679	0.685	0.565	0.726
9	1.000	0.793	1.000	0.935	0.609	0.669	0.645	0.651	0.669

*Refer to Subsection IV.D.3.

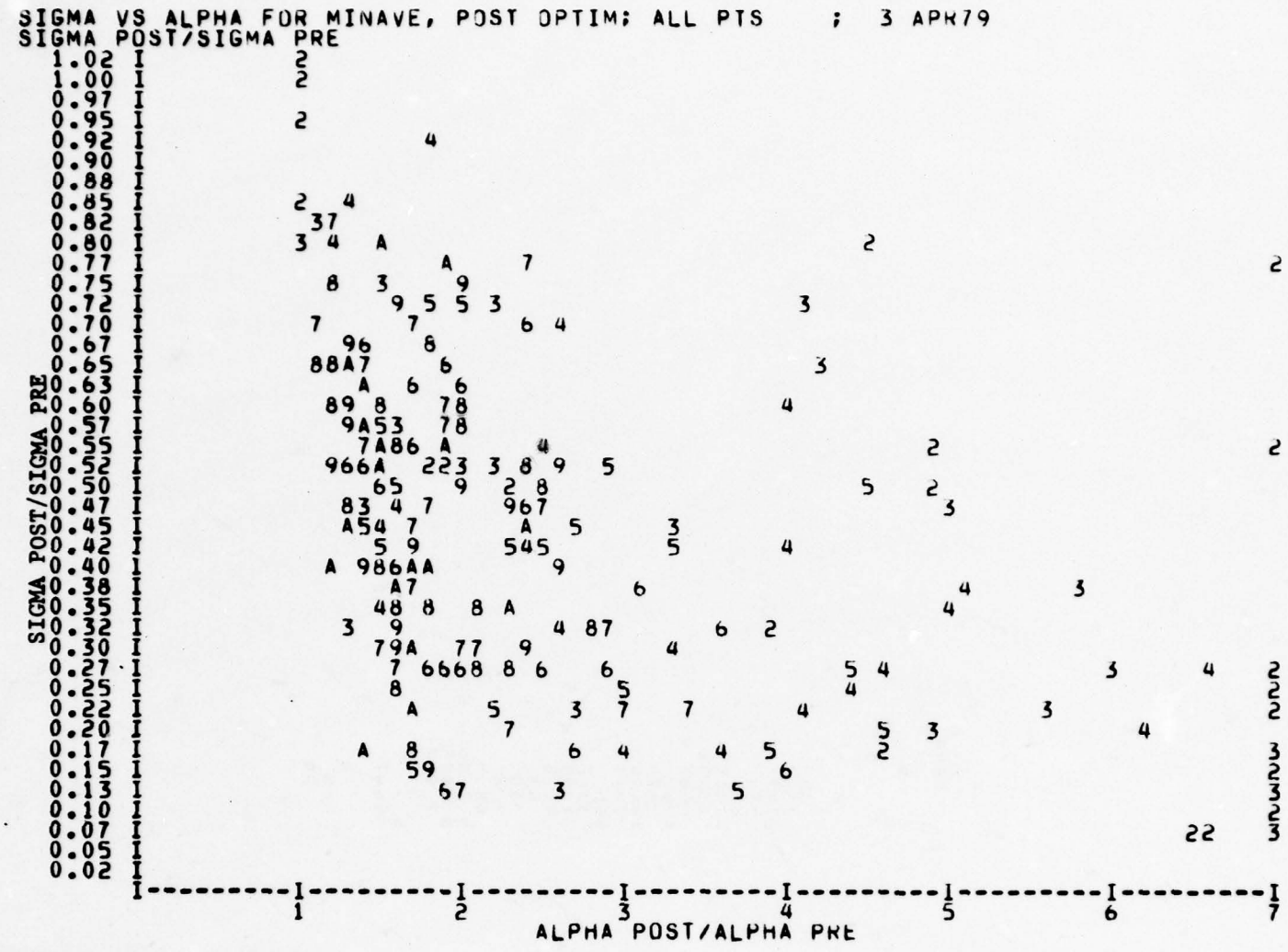


Figure 28. $\sigma_{post}/\sigma_{pre}$ Versus $\alpha_{post}/\alpha_{pre}$ for MINAVE Agglomerative Clustering of Scanning Patterns

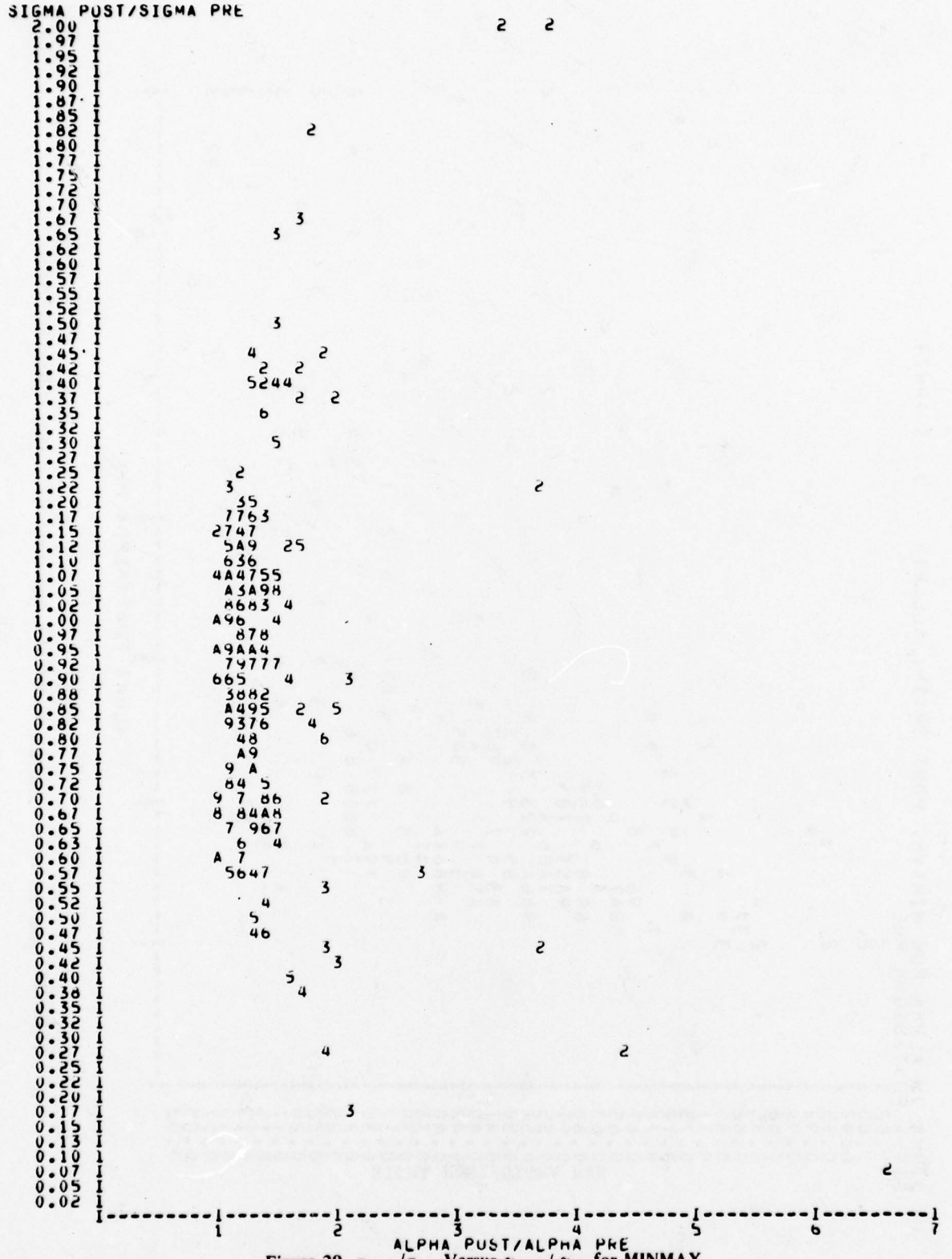


Figure 29. $\sigma_{post}/\sigma_{pre}$ Versus $\alpha_{post}/\alpha_{pre}$ for MINMAX Agglomerative Clustering of Scanning Patterns

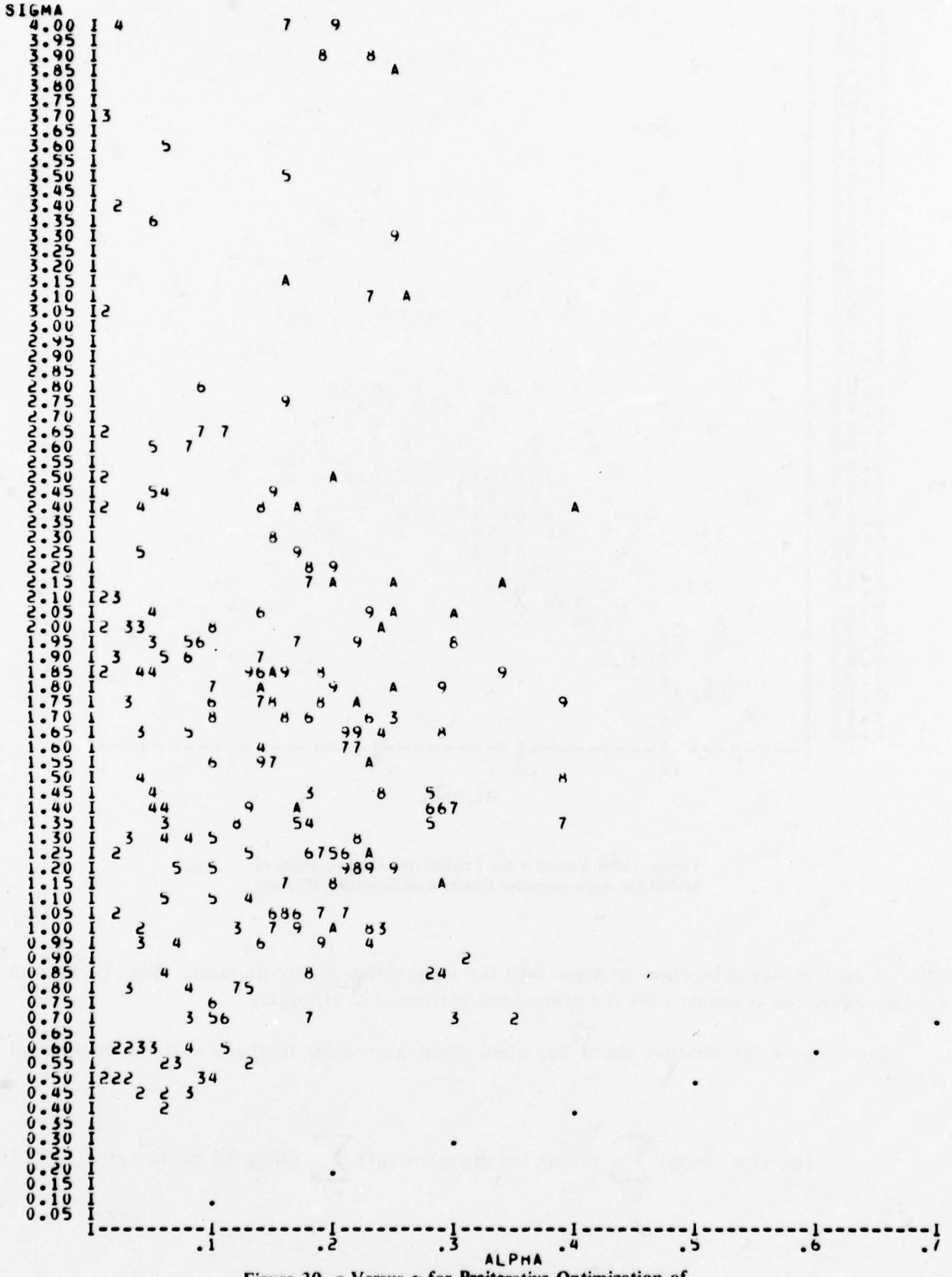


Figure 30. σ Versus α for Preiterative Optimization of MINAVE Agglomerative Clusters of Scanning Patterns

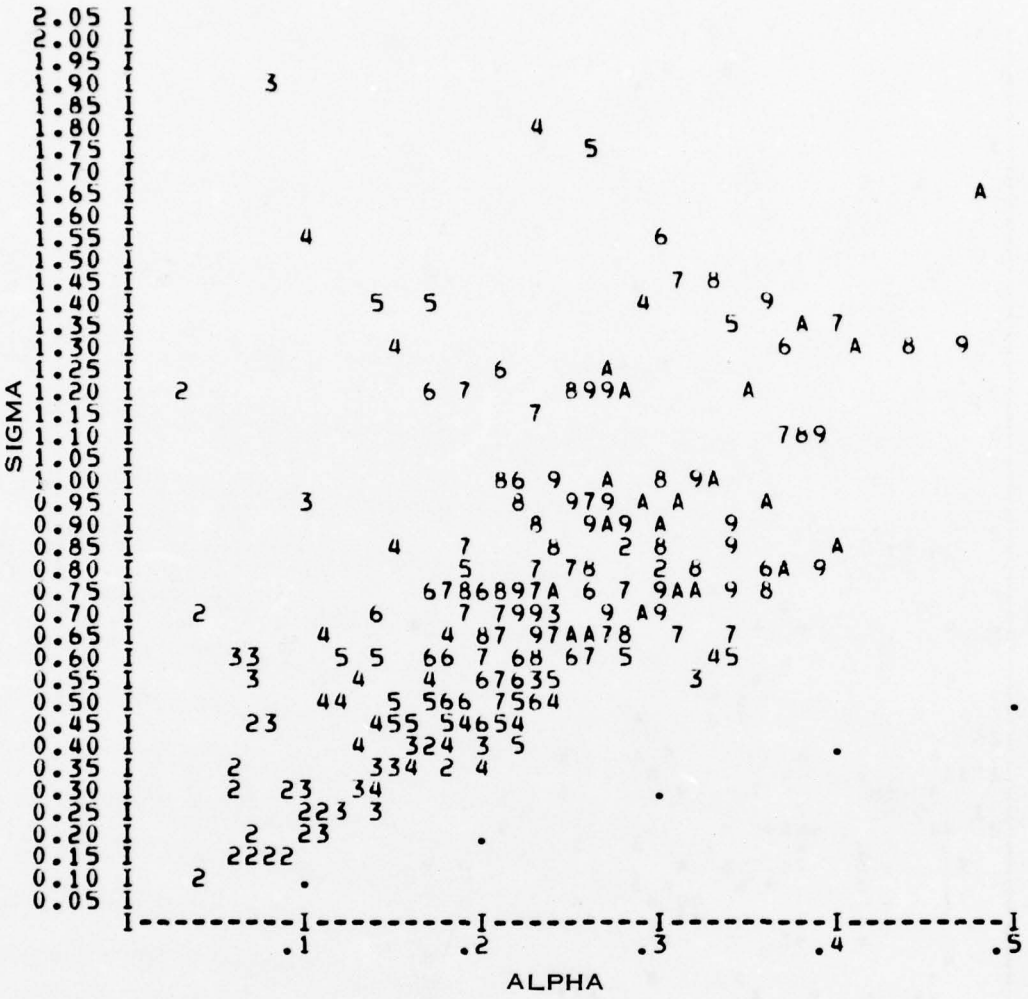


Figure 31. σ Versus α for Preiterative Optimization of MINMAX Agglomerative Clusters of Scanning Patterns

("H" is used in this subsection to agree with the information theory literature. The "E" used in the last subsection is reserved for the normalized entropy, i.e., $H/\log_2 c$.)

The average information about the class, given knowledge of the sex, is the conditional entropy:

$$H(c|s) = -p(m) \sum_{i=1}^{c} p(i|m) \log p(i|m) - p(f) \sum_{i=1}^{c} p(i|f) \log p(i|f) \qquad (22)$$

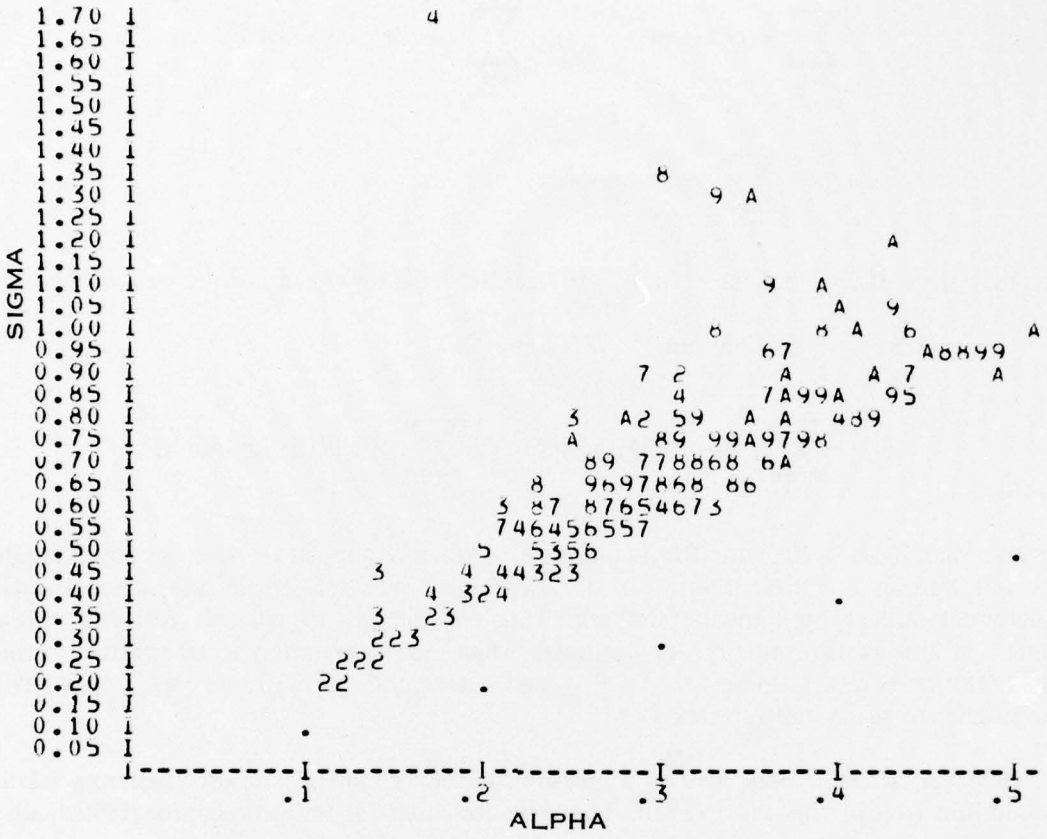


Figure 32. σ Versus α for Postiterative Optimization of MINAVE Agglomerative Clusters of Scanning Patterns

The average information gained from the clustering is then reduced by the *a priori* information represented in the classes, yielding what information theorists term I (c;s), the average of the mutual information between the class and the sex. The value for I(c;s) is given by

$$I(c;s) = H(c) - H(c|s)$$

$$= -\sum_{i=1}^{c} p(i) \log p(i) + p(m) \sum_{i=1}^{c} p(i|m) \log p(i|m) \qquad (23)$$

$$+ p(f) \sum_{i=1}^{c} p(i|f) \log p(i|f)$$

Note that since $H(c) - H(c|s) = H(s) - H(s|c)$, the term I(c;s) can also be written as:

$$I(c;s) = - p(m) \log p(m) - p(f) \log p(f) \qquad (24)$$

$$+ \sum_{i=1}^{c} p(i)\, p(m|i) \log p(m|i) + \sum_{i=1}^{c} p(i)\, p(f|i) \log p(m|i)$$

Note also that I(c;s) = H(c) for the two-class case when a population with an equal number of males and females are divided into an all-male class and an all-female class since the class is uniquely determined by knowing the sex. This corresponds to what is called a "noise-free channel" in information theory. In contrast, when no information is transmitted through a channel, H(c) = H(c|s), yielding I (c;s) = 0, which corresponds to the case of each class having an equal number of males and females.

However, I(c;s) is used here as a measure of the "goodness" of the clustering relative to the condition (sex in this case) tested. The estimates used for the various probabilities are given in terms of

n	Total number of samples
n_i	Number of samples in class i
n^m	Number of males
n^f	Number of females
n_i^m	Number of males in class i
n_i^f	Number of females in class i

The given probabilities are estimated as

$p(i) = n_i/n$

$p(m) = n^m/n$

$p(f) = n^f/n$

$p(i|m) = n_i^m/n^m$

$p(i|f) = n_i^f/n^f$

TABLE 18. MARTIN-HERSCHER DIGIT TEXTS

$$p(m|i) = n_i^m/n_i$$

$$p(f|i) = n_i^f/n_i$$

Hence, I(c;s) is calculated by

$$I(c;s) = -\sum_{i=1}^{c} (n_i/n) \log (n_i/n) + (n^m/n) \sum_{i=1}^{c} (n_i^m/n^m) \log (n_i^m/n^m) + (n^f/n) \sum_{i=1}^{c} (n_i^f/n^f) \log (n_i^f/n^f) \quad (25)$$

Values of I are given in Tables 9 through 12 for scanning patterns and in Tables 13 through 16 for recognition patterns, along with another measure, R, the distribution of males and females among the classes, for the reader with a less esoteric inclination. The value for R is given by

$$R = \frac{1}{n} \sum_{i=1}^{c} \min (n_i^f, n_i^m) \quad (26)$$

which is a measure of the residue in each of the classes. The tables in each of the two sets are arranged by clustering algorithm in the following order:

(1) MINAVE agglomerative clustering; preiterative optimization

(2) MINAVE agglomerative clustering; postiterative optimization

(3) MINMAX agglomerative clustering; preiterative optimization

(4) MINMAX agglomerative clustering; postiterative optimization.

The information in Tables 9 through 12 is summarized in Figure 33 with histograms of I for each of cases 1 through 4 above. Likewise, the information in Tables 13 through 16 is summarized in Figure 34. It is clear from these two figures that iterative optimization to minimize J_e improves the resulting clusters, assuming the male/female distinction is valid (and from an acoustic-phonetic standpoint, it is). In addition, these two figures show that MINMAX agglomerative clustering yields better clusters before the iterative optimization than does MINAVE. However, no clear preference results between the MINAVE and MINMAX clusters after iterative optimization, showing that the iterative optimization algorithm was robust enough to produce good clusters from either the MINAVE or MINMAX agglomerative clustering, even though the starting partition produced from the MINAVE agglomeration was clearly inferior.

TABLE 19. SYNOPSIS OF EVALUATION RESULTS

	Percent

TABLE 9. MUTUAL INFORMATION AND RESIDUES FOR PREITERATIVE OPTIMIZATION OF MINAVE AGGLOMERATIVE CLUSTERS OF SCANNING PATTERNS

DG/RP	C:	2	3	4	5	6	7	8	9	10
0 1	I:	0.030	0.091	0.119	0.179	0.195	0.217	0.288	0.289	0.289
0 1	R:	0.434	0.410	0.410	0.355	0.355	0.349	0.259	0.259	0.259
0 2	I:	0.006	0.024	0.036	0.038	0.372	0.372	0.395	0.401	0.466
0 2	R:	0.494	0.482	0.482	0.470	0.187	0.187	0.175	0.175	0.159
0 3	I:	0.052	0.093	0.103	0.103	0.125	0.125	0.130	0.143	0.143
0 3	R:	0.392	0.373	0.373	0.373	0.355	0.355	0.355	0.355	0.355
1 1	I:	0.006	0.014	0.015	0.034	0.043	0.049	0.062	0.075	0.075
1 1	R:	0.488	0.488	0.482	0.435	0.435	0.435	0.435	0.423	0.423
1 2	I:	0.006	0.012	0.082	0.086	0.086	0.086	0.097	0.097	0.124
1 2	R:	0.488	0.488	0.423	0.411	0.411	0.411	0.411	0.411	0.387
2 1	I:	0.005	0.006	0.026	0.032	0.038	0.039	0.058	0.064	0.071
2 1	R:	0.488	0.482	0.476	0.470	0.464	0.458	0.411	0.411	0.405
2 2	I:	0.018	0.018	0.018	0.025	0.714	0.755	0.755	0.757	0.794
2 2	R:	0.482	0.482	0.482	0.476	0.060	0.042	0.042	0.042	0.036
2 3	I:	0.007	0.025	0.026	0.030	0.042	0.062	0.062	0.069	0.069
2 3	R:	0.452	0.429	0.429	0.429	0.429	0.411	0.411	0.411	0.411
3 1	I:	0.000	0.028	0.035	0.040	0.046	0.050	0.054	0.079	0.081
3 1	R:	0.497	0.456	0.450	0.444	0.420	0.420	0.420	0.408	0.402
3 2	I:	0.000	0.007	0.046	0.046	0.074	0.154	0.154	0.162	0.165
3 2	R:	0.497	0.491	0.438	0.438	0.432	0.320	0.320	0.314	0.314
4 1	I:	0.006	0.049	0.061	0.087	0.118	0.124	0.126	0.133	0.138
4 1	R:	0.497	0.455	0.455	0.443	0.413	0.413	0.413	0.389	0.389
4 2	I:	0.006	0.018	0.018	0.124	0.140	0.146	0.208	0.265	0.320
4 2	R:	0.497	0.485	0.485	0.377	0.347	0.347	0.335	0.269	0.251
5 1	I:	0.055	0.146	0.218	0.222	0.230	0.236	0.236	0.245	0.245
5 1	R:	0.432	0.355	0.302	0.302	0.302	0.302	0.302	0.302	0.302
5 2	I:	0.000	0.006	0.043	0.049	0.296	0.314	0.332	0.356	0.358
5 2	R:	0.497	0.491	0.456	0.456	0.201	0.201	0.201	0.201	0.201
6 1	I:	0.206	0.230	0.230	0.268	0.272	0.620	0.621	0.642	0.659
6 1	R:	0.281	0.275	0.275	0.275	0.275	0.078	0.078	0.072	0.066
6 2	I:	0.006	0.012	0.018	0.633	0.634	0.634	0.714	0.714	0.720
6 2	R:	0.497	0.491	0.485	0.072	0.072	0.072	0.072	0.072	0.072
6 3	I:	0.005	0.017	0.051	0.178	0.185	0.207	0.207	0.248	0.248
6 3	R:	0.491	0.479	0.419	0.317	0.317	0.317	0.317	0.287	0.287
7 1	I:	0.023	0.029	0.570	0.598	0.620	0.623	0.623	0.641	0.648
7 1	R:	0.488	0.482	0.089	0.089	0.089	0.089	0.089	0.089	0.083
7 2	I:	0.006	0.591	0.596	0.601	0.603	0.604	0.604	0.604	0.604
7 2	R:	0.488	0.083	0.083	0.083	0.083	0.083	0.083	0.083	0.083
7 3	I:	0.006	0.031	0.036	0.081	0.101	0.101	0.120	0.120	0.141
7 3	R:	0.488	0.464	0.452	0.452	0.446	0.446	0.399	0.399	0.399
8 1	I:	0.041	0.104	0.111	0.139	0.405	0.421	0.475	0.476	0.478
8 1	R:	0.435	0.381	0.375	0.375	0.161	0.161	0.149	0.149	0.149
8 2	I:	0.008	0.082	0.103	0.113	0.126	0.126	0.126	0.126	0.144
8 2	R:	0.488	0.405	0.405	0.405	0.393	0.393	0.393	0.393	0.357
9 1	I:	0.023	0.046	0.050	0.050	0.270	0.285	0.306	0.334	0.353
9 1	R:	0.485	0.456	0.456	0.456	0.219	0.219	0.207	0.207	0.207
9 2	I:	0.012	0.021	0.032	0.034	0.040	0.082	0.085	0.099	0.247
9 2	R:	0.491	0.473	0.473	0.467	0.467	0.402	0.402	0.396	0.296

TABLE 10. MUTUAL INFORMATION AND RESIDUES FOR POSTITERATIVE OPTIMIZATION OF MINAVE AGGLOMERATIVE CLUSTERS OF SCANNING PATTERNS

DG/RP	C:	2	3	4	5	6	7	8	9	10
0 1	I:	0.034	0.575	0.387	0.524	0.588	0.552	0.598	0.621	0.631
0 1	R:	0.422	0.096	0.211	0.151	0.096	0.145	0.114	0.108	0.102
0 2	I:	0.389	0.447	0.404	0.477	0.493	0.560	0.556	0.603	0.578
0 2	R:	0.157	0.181	0.217	0.139	0.151	0.157	0.145	0.151	0.139
0 3	I:	0.049	0.067	0.177	0.245	0.286	0.215	0.257	0.269	0.276
0 3	R:	0.392	0.366	0.265	0.253	0.253	0.307	0.271	0.301	0.289
1 1	I:	0.023	0.015	0.040	0.048	0.059	0.046	0.051	0.079	0.102
1 1	R:	0.411	0.429	0.405	0.399	0.387	0.393	0.393	0.363	0.381
1 2	I:	0.323	0.285	0.455	0.428	0.420	0.382	0.464	0.401	0.416
1 2	R:	0.190	0.226	0.185	0.190	0.179	0.185	0.202	0.214	0.202
2 1	I:	0.075	0.063	0.089	0.080	0.129	0.126	0.132	0.134	0.188
2 1	R:	0.357	0.375	0.351	0.363	0.310	0.315	0.304	0.304	0.280
2 2	I:	0.674	0.658	0.569	0.705	0.712	0.759	0.732	0.665	0.704
2 2	R:	0.060	0.077	0.101	0.060	0.083	0.083	0.089	0.107	0.089
2 3	I:	0.013	0.013	0.014	0.069	0.056	0.069	0.065	0.113	0.155
2 3	R:	0.435	0.435	0.446	0.357	0.375	0.363	0.369	0.345	0.333
3 1	I:	0.009	0.029	0.045	0.079	0.057	0.131	0.180	0.169	0.158
3 1	R:	0.444	0.402	0.402	0.355	0.373	0.337	0.308	0.325	0.302
3 2	I:	0.058	0.062	0.103	0.159	0.141	0.157	0.181	0.281	0.289
3 2	R:	0.361	0.361	0.337	0.302	0.320	0.320	0.314	0.243	0.237
4 1	I:	0.001	0.036	0.306	0.296	0.309	0.286	0.258	0.402	0.355
4 1	R:	0.485	0.401	0.228	0.216	0.198	0.234	0.269	0.192	0.234
4 2	I:	0.220	0.341	0.343	0.256	0.316	0.342	0.395	0.363	0.435
4 2	R:	0.234	0.198	0.198	0.299	0.234	0.263	0.180	0.204	0.186
5 1	I:	0.523	0.435	0.397	0.522	0.610	0.617	0.451	0.557	0.535
5 1	R:	0.112	0.136	0.166	0.130	0.118	0.130	0.189	0.142	0.130
5 2	I:	0.435	0.381	0.416	0.329	0.429	0.378	0.424	0.400	0.428
5 2	R:	0.136	0.172	0.178	0.231	0.189	0.189	0.195	0.189	0.183
6 1	I:	0.235	0.451	0.426	0.614	0.600	0.636	0.723	0.671	0.722
6 1	R:	0.269	0.162	0.156	0.078	0.084	0.078	0.066	0.084	0.072
6 2	I:	0.749	0.761	0.799	0.740	0.668	0.633	0.732	0.603	0.729
6 2	R:	0.042	0.042	0.036	0.048	0.090	0.138	0.108	0.144	0.090
6 3	I:	0.177	0.236	0.259	0.346	0.304	0.413	0.413	0.345	0.343
6 3	R:	0.269	0.257	0.263	0.186	0.210	0.192	0.174	0.198	0.198
7 1	I:	0.732	0.599	0.701	0.683	0.728	0.655	0.662	0.650	0.764
7 1	R:	0.048	0.089	0.065	0.077	0.065	0.089	0.083	0.101	0.065
7 2	I:	0.607	0.685	0.573	0.720	0.680	0.717	0.709	0.736	0.680
7 2	R:	0.077	0.071	0.095	0.065	0.095	0.071	0.071	0.077	0.101
7 3	I:	0.014	0.335	0.346	0.313	0.375	0.425	0.404	0.388	0.445
7 3	R:	0.435	0.238	0.220	0.214	0.232	0.208	0.226	0.220	0.226
8 1	I:	0.139	0.281	0.183	0.300	0.299	0.322	0.461	0.417	0.427
8 1	R:	0.286	0.214	0.286	0.202	0.208	0.196	0.167	0.173	0.190
8 2	I:	0.334	0.238	0.289	0.575	0.392	0.528	0.445	0.496	0.527
8 2	R:	0.185	0.250	0.220	0.137	0.208	0.167	0.250	0.185	0.196
9 1	I:	0.494	0.515	0.484	0.456	0.544	0.514	0.557	0.566	0.568
9 1	R:	0.112	0.154	0.225	0.207	0.142	0.154	0.136	0.124	0.124
9 2	I:	0.015	0.121	0.296	0.292	0.252	0.263	0.281	0.291	0.333
9 2	R:	0.432	0.308	0.237	0.249	0.231	0.254	0.249	0.231	0.219

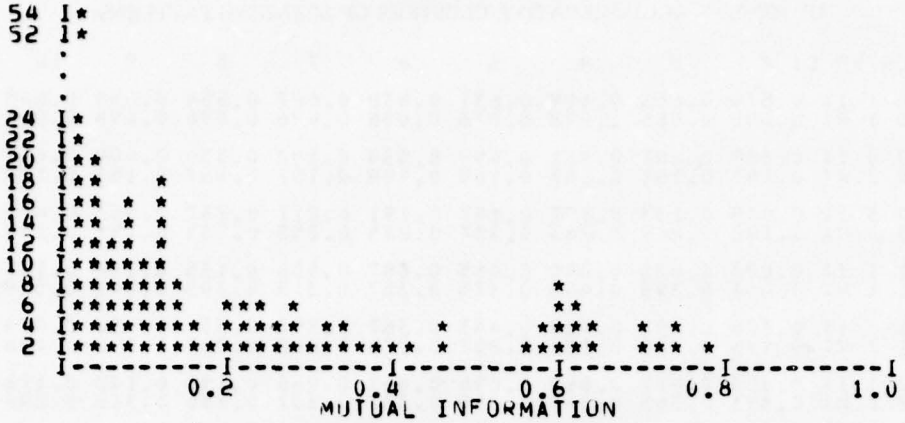
TABLE 11. MUTUAL INFORMATION AND RESIDUES FOR PREITERATIVE OPTIMIZATION OF MINMAX AGGLOMERATIVE CLUSTERS OF SCANNING PATTERNS

DG/RP	C:	2	3	4	5	6	7	8	9	10
0 1	I:	0.222	0.356	0.398	0.398	0.399	0.423	0.423	0.438	0.440
0 1	R:	0.235	0.169	0.169	0.169	0.169	0.169	0.169	0.169	0.169
0 2	I:	0.022	0.105	0.133	0.373	0.390	0.453	0.461	0.461	0.466
0 2	R:	0.416	0.380	0.380	0.205	0.199	0.163	0.163	0.163	0.157
0 3	I:	0.052	0.060	0.072	0.079	0.165	0.182	0.277	0.291	0.291
0 3	R:	0.392	0.392	0.386	0.386	0.271	0.271	0.235	0.235	0.235
1 1	I:	0.009	0.017	0.032	0.035	0.092	0.119	0.120	0.130	0.134
1 1	R:	0.464	0.429	0.417	0.417	0.393	0.393	0.393	0.393	0.393
1 2	I:	0.031	0.032	0.155	0.158	0.159	0.214	0.223	0.525	0.525
1 2	R:	0.458	0.458	0.310	0.310	0.304	0.304	0.304	0.149	0.149
2 1	I:	0.005	0.006	0.065	0.080	0.088	0.104	0.104	0.104	0.126
2 1	R:	0.476	0.476	0.363	0.357	0.357	0.357	0.357	0.357	0.345
2 2	I:	0.005	0.005	0.343	0.455	0.493	0.516	0.522	0.528	0.528
2 2	R:	0.464	0.464	0.238	0.238	0.196	0.185	0.179	0.179	0.179
2 3	I:	0.002	0.006	0.033	0.034	0.041	0.054	0.056	0.057	0.063
2 3	R:	0.462	0.458	0.429	0.429	0.429	0.417	0.411	0.411	0.411
3 1	I:	0.004	0.078	0.090	0.173	0.174	0.203	0.204	0.204	0.204
3 1	R:	0.467	0.367	0.349	0.308	0.308	0.308	0.308	0.308	0.308
3 2	I:	0.093	0.133	0.169	0.169	0.199	0.207	0.212	0.221	0.230
3 2	R:	0.331	0.331	0.272	0.272	0.272	0.272	0.272	0.272	0.272
4 1	I:	0.000	0.026	0.186	0.239	0.311	0.350	0.361	0.363	0.370
4 1	R:	0.497	0.473	0.293	0.246	0.210	0.198	0.198	0.198	0.198
4 2	I:	0.006	0.013	0.062	0.084	0.099	0.130	0.173	0.405	0.420
4 2	R:	0.497	0.449	0.395	0.395	0.359	0.359	0.329	0.216	0.216
5 1	I:	0.001	0.083	0.144	0.167	0.304	0.340	0.382	0.385	0.418
5 1	R:	0.479	0.396	0.361	0.320	0.254	0.231	0.219	0.219	0.219
5 2	I:	0.065	0.266	0.290	0.375	0.387	0.414	0.470	0.483	0.483
5 2	R:	0.355	0.237	0.237	0.189	0.189	0.189	0.189	0.189	0.189
6 1	I:	0.250	0.330	0.352	0.556	0.558	0.569	0.569	0.570	0.592
6 1	R:	0.251	0.251	0.246	0.096	0.096	0.096	0.096	0.096	0.090
6 2	I:	0.550	0.556	0.619	0.621	0.622	0.691	0.697	0.712	0.715
6 2	R:	0.102	0.102	0.102	0.102	0.102	0.072	0.072	0.072	0.072
6 3	I:	0.111	0.169	0.201	0.212	0.230	0.234	0.240	0.247	0.265
6 3	R:	0.323	0.269	0.246	0.246	0.246	0.246	0.246	0.246	0.228
7 1	I:	0.017	0.185	0.213	0.260	0.285	0.322	0.323	0.444	0.455
7 1	R:	0.488	0.298	0.286	0.286	0.286	0.262	0.262	0.190	0.190
7 2	I:	0.008	0.523	0.524	0.534	0.582	0.610	0.631	0.655	0.655
7 2	R:	0.458	0.119	0.119	0.119	0.119	0.119	0.119	0.119	0.119
7 3	I:	0.198	0.246	0.251	0.251	0.262	0.266	0.266	0.279	0.295
7 3	R:	0.321	0.321	0.321	0.321	0.292	0.292	0.292	0.292	0.292
8 1	I:	0.081	0.129	0.136	0.142	0.143	0.191	0.261	0.267	0.287
8 1	R:	0.351	0.304	0.304	0.304	0.304	0.304	0.268	0.268	0.262
8 2	I:	0.095	0.200	0.217	0.224	0.224	0.231	0.487	0.496	0.505
8 2	R:	0.327	0.292	0.280	0.280	0.280	0.280	0.185	0.179	0.179
9 1	I:	0.220	0.236	0.271	0.380	0.396	0.523	0.548	0.548	0.553
9 1	R:	0.231	0.231	0.231	0.231	0.231	0.160	0.148	0.148	0.148
9 2	I:	0.059	0.063	0.080	0.113	0.285	0.285	0.285	0.294	0.328
9 2	R:	0.361	0.361	0.361	0.361	0.272	0.272	0.272	0.272	0.249

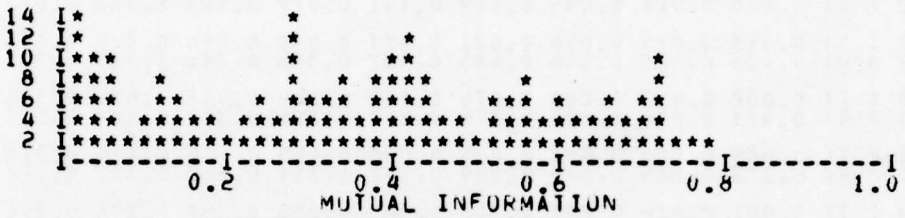
$$\left[\text{TE}_k/\text{no. of column in digit k}\right] + \frac{w_k}{\sum_{\text{PPE}}^{\text{NPP}}} \tag{29}$$

TABLE 12. MUTUAL INFORMATION AND RESIDUES FOR POSTITERATIVE OPTIMIZATION OF MINMAX AGGLOMERATIVE CLUSTERS OF SCANNING PATTERNS

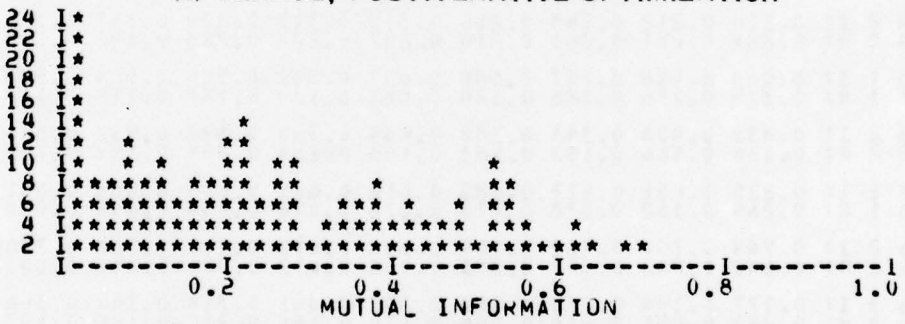
DG/RP	C:	2	3	4	5	6	7	8	9	10
0 1	I:	0.574	0.666	0.609	0.631	0.630	0.627	0.654	0.659	0.665
0 1	R:	0.090	0.066	0.078	0.078	0.096	0.096	0.096	0.096	0.096
0 2	I:	0.389	0.447	0.411	0.492	0.534	0.542	0.536	0.604	0.645
0 2	R:	0.157	0.181	0.187	0.169	0.199	0.157	0.157	0.151	0.120
0 3	I:	0.049	0.133	0.177	0.147	0.191	0.217	0.287	0.353	0.403
0 3	R:	0.392	0.289	0.265	0.337	0.265	0.253	0.235	0.199	0.169
1 1	I:	0.023	0.035	0.040	0.065	0.087	0.156	0.135	0.130	0.137
1 1	R:	0.411	0.393	0.405	0.375	0.357	0.315	0.345	0.345	0.339
1 2	I:	0.328	0.290	0.459	0.443	0.367	0.398	0.339	0.452	0.424
1 2	R:	0.196	0.226	0.185	0.202	0.226	0.202	0.268	0.196	0.196
2 1	I:	0.053	0.077	0.062	0.096	0.061	0.066	0.190	0.148	0.176
2 1	R:	0.393	0.363	0.375	0.369	0.387	0.387	0.286	0.315	0.292
2 2	I:	0.674	0.658	0.711	0.617	0.684	0.792	0.698	0.709	0.751
2 2	R:	0.060	0.077	0.095	0.119	0.101	0.077	0.101	0.083	0.071
2 3	I:	0.013	0.019	0.014	0.021	0.077	0.093	0.096	0.128	0.145
2 3	R:	0.435	0.429	0.446	0.435	0.387	0.345	0.345	0.327	0.327
3 1	I:	0.002	0.012	0.023	0.076	0.066	0.100	0.082	0.074	0.143
3 1	R:	0.473	0.444	0.426	0.379	0.367	0.355	0.373	0.361	0.343
3 2	I:	0.049	0.108	0.093	0.095	0.095	0.158	0.175	0.179	0.214
3 2	R:	0.373	0.325	0.349	0.349	0.343	0.314	0.302	0.302	0.278
4 1	I:	0.001	0.036	0.287	0.305	0.265	0.255	0.298	0.294	0.316
4 1	R:	0.485	0.401	0.222	0.216	0.269	0.275	0.257	0.269	0.263
4 2	I:	0.220	0.212	0.263	0.286	0.315	0.347	0.329	0.367	0.376
4 2	R:	0.234	0.251	0.263	0.216	0.257	0.222	0.228	0.192	0.198
5 1	I:	0.044	0.414	0.457	0.540	0.627	0.552	0.546	0.589	0.554
5 1	R:	0.379	0.178	0.148	0.124	0.083	0.130	0.124	0.116	0.118
5 2	I:	0.435	0.408	0.393	0.380	0.439	0.364	0.397	0.532	0.536
5 2	R:	0.136	0.166	0.183	0.183	0.148	0.169	0.207	0.154	0.154
6 1	I:	0.235	0.451	0.619	0.625	0.612	0.611	0.600	0.558	0.607
6 1	R:	0.269	0.162	0.078	0.078	0.078	0.078	0.084	0.096	0.084
6 2	I:	0.749	0.754	0.762	0.789	0.694	0.676	0.705	0.760	0.790
6 2	R:	0.042	0.042	0.042	0.048	0.066	0.072	0.066	0.054	0.048
6 3	I:	0.177	0.194	0.285	0.346	0.390	0.381	0.314	0.344	0.368
6 3	R:	0.269	0.251	0.210	0.186	0.174	0.186	0.216	0.198	0.192
7 1	I:	0.732	0.601	0.624	0.611	0.488	0.496	0.544	0.781	0.778
7 1	R:	0.048	0.089	0.089	0.089	0.149	0.149	0.143	0.048	0.054
7 2	I:	0.607	0.487	0.661	0.665	0.684	0.684	0.747	0.723	0.723
7 2	R:	0.077	0.131	0.077	0.095	0.095	0.095	0.095	0.089	0.089
7 3	I:	0.446	0.332	0.347	0.363	0.410	0.401	0.428	0.435	0.490
7 3	R:	0.149	0.238	0.220	0.214	0.185	0.185	0.179	0.214	0.161
8 1	I:	0.139	0.185	0.236	0.248	0.235	0.215	0.274	0.291	0.349
8 1	R:	0.286	0.274	0.274	0.268	0.280	0.268	0.250	0.250	0.226
8 2	I:	0.312	0.452	0.366	0.441	0.519	0.450	0.542	0.575	0.593
8 2	R:	0.196	0.190	0.244	0.196	0.155	0.196	0.179	0.155	0.149
9 1	I:	0.494	0.449	0.534	0.551	0.547	0.512	0.563	0.567	0.585
9 1	R:	0.112	0.130	0.136	0.136	0.136	0.142	0.118	0.118	0.112
9 2	I:	0.110	0.030	0.240	0.292	0.288	0.270	0.291	0.302	0.367
9 2	R:	0.320	0.426	0.278	0.249	0.284	0.284	0.278	0.243	0.201



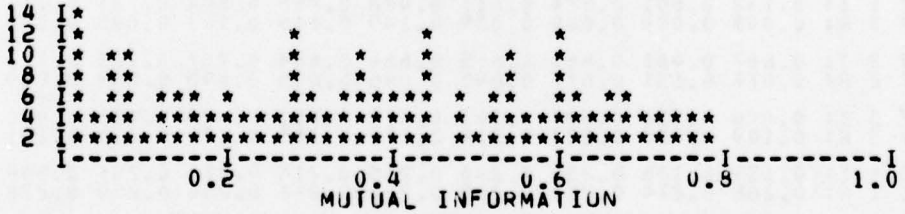
(A) MINAVE, PREITERATIVE OPTIMIZATION



(B) MINAVE, POSTITERATIVE OPTIMIZATION



(C) MINMAX, PREITERATIVE OPTIMIZATION



(D) MINMAX, POSTITERATIVE OPTIMIZATION

Figure 33. Comparison of Mutual Information for Clustered Scanning Patterns Using Four Clustering Algorithms

TABLE 21. T-FUNCTION QUANTIZATION THRESHOLDS

TABLE 13. MUTUAL INFORMATION AND RESIDUES FOR PREITERATIVE OPTIMIZATION OF MINAVE AGGLOMERATIVE CLUSTERS OF RECOGNITION PATTERNS

DG/RP	C:	2	3	4	5	6	7	8	9	10
0 0	I:	0.024	0.088	0.088	0.088	0.095	0.095	0.095	0.095	0.693
0 0	R:	0.482	0.422	0.422	0.422	0.410	0.410	0.410	0.410	0.066
1 0	I:	0.403	0.403	0.447	0.448	0.470	0.484	0.497	0.500	0.500
1 0	R:	0.167	0.167	0.143	0.143	0.137	0.131	0.131	0.131	0.131
2 0	I:	0.024	0.043	0.049	0.061	0.727	0.729	0.729	0.740	0.741
2 0	R:	0.476	0.458	0.458	0.446	0.060	0.060	0.060	0.060	0.060
3 0	I:	0.006	0.014	0.023	0.541	0.567	0.568	0.572	0.575	0.577
3 0	R:	0.497	0.479	0.462	0.101	0.101	0.101	0.101	0.101	0.101
4 0	I:	0.006	0.018	0.024	0.197	0.198	0.196	0.235	0.251	0.278
4 0	R:	0.497	0.485	0.479	0.311	0.311	0.311	0.305	0.287	0.269
5 0	I:	0.661	0.708	0.736	0.752	0.767	0.768	0.768	0.784	0.797
5 0	R:	0.065	0.059	0.053	0.047	0.047	0.047	0.047	0.041	0.036
6 0	I:	0.012	0.631	0.632	0.634	0.637	0.637	0.640	0.640	0.668
6 0	R:	0.491	0.078	0.078	0.078	0.078	0.078	0.078	0.078	0.078
7 0	I:	0.023	0.789	0.790	0.791	0.791	0.793	0.793	0.814	0.814
7 0	R:	0.488	0.042	0.042	0.042	0.042	0.042	0.042	0.042	0.042
8 0	I:	0.006	0.023	0.872	0.873	0.873	0.874	0.874	0.875	0.881
8 0	R:	0.488	0.488	0.018	0.018	0.018	0.018	0.018	0.018	0.018
9 0	I:	0.031	0.119	0.128	0.128	0.136	0.170	0.170	0.391	0.393
9 0	R:	0.462	0.385	0.385	0.385	0.367	0.367	0.367	0.183	0.183

TABLE 14. MUTUAL INFORMATION AND RESIDUES FOR POSTITERATIVE OPTIMIZATION OF MINAVE AGGLOMERATIVE CLUSTERS OF RECOGNITION PATTERNS

DG/RP	C:	2	3	4	5	6	7	8	9	10
0 0	I:	0.787	0.678	0.685	0.688	0.656	0.810	0.743	0.696	0.788
0 0	R:	0.042	0.060	0.060	0.066	0.072	0.042	0.054	0.060	0.054
1 0	I:	0.341	0.318	0.434	0.447	0.446	0.375	0.560	0.501	0.490
1 0	R:	0.173	0.202	0.167	0.179	0.155	0.179	0.107	0.143	0.149
2 0	I:	0.626	0.515	0.670	0.676	0.659	0.631	0.768	0.717	0.800
2 0	R:	0.083	0.131	0.083	0.071	0.077	0.125	0.054	0.077	0.054
3 0	I:	0.401	0.347	0.404	0.483	0.421	0.561	0.612	0.623	0.627
3 0	R:	0.154	0.219	0.148	0.160	0.160	0.124	0.130	0.146	0.142
4 0	I:	0.151	0.165	0.213	0.298	0.398	0.396	0.322	0.325	0.432
4 0	R:	0.281	0.311	0.311	0.234	0.180	0.204	0.240	0.240	0.204
5 0	I:	0.549	0.583	0.829	0.795	0.706	0.724	0.725	0.700	0.711
5 0	R:	0.095	0.142	0.083	0.089	0.118	0.083	0.065	0.083	0.065
6 0	I:	0.841	0.769	0.682	0.728	0.734	0.739	0.785	0.811	0.813
6 0	R:	0.024	0.048	0.084	0.054	0.060	0.054	0.042	0.054	0.054
7 0	I:	0.953	0.829	0.819	0.768	0.848	0.852	0.933	0.785	0.791
7 0	R:	0.006	0.030	0.077	0.065	0.036	0.036	0.012	0.071	0.065
8 0	I:	0.703	0.797	0.840	0.840	0.832	0.764	0.795	0.802	0.887
8 0	R:	0.054	0.036	0.030	0.030	0.042	0.083	0.077	0.077	0.024
9 0	I:	0.554	0.642	0.505	0.725	0.750	0.636	0.664	0.739	0.783
9 0	R:	0.095	0.077	0.154	0.059	0.089	0.124	0.112	0.071	0.077

TABLE 24. RECOGNITION ERROR (TE) NORMALIZING CONSTANTS

Digit	Source 1	Source 2	Source 3	Source 4	Source 5

TABLE 15. MUTUAL INFORMATION AND RESIDUES FOR PREITERATIVE OPTIMIZATION OF MINMAX AGGLOMERATIVE CLUSTERS OF RECOGNITION PATTERNS

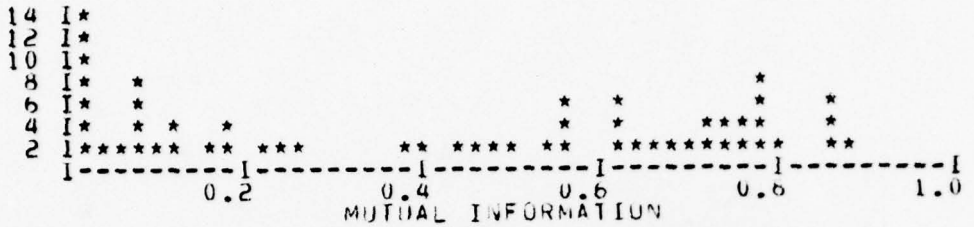
DG/RP	C:	2	3	4	5	6	7	8	9	10
0 0	I:	0.212	0.707	0.711	0.716	0.721	0.721	0.724	0.740	0.776
0 0	R:	0.301	0.054	0.054	0.054	0.054	0.054	0.054	0.054	0.054
1 0	I:	0.176	0.222	0.349	0.355	0.355	0.396	0.433	0.433	0.446
1 0	R:	0.262	0.262	0.202	0.202	0.202	0.202	0.179	0.179	0.179
2 0	I:	0.094	0.226	0.496	0.525	0.549	0.549	0.549	0.549	0.549
2 0	R:	0.321	0.250	0.149	0.149	0.149	0.149	0.149	0.149	0.149
3 0	I:	0.325	0.326	0.356	0.407	0.428	0.473	0.510	0.538	0.538
3 0	R:	0.195	0.195	0.195	0.183	0.183	0.178	0.130	0.130	0.130
4 0	I:	0.162	0.170	0.204	0.205	0.205	0.230	0.250	0.250	0.250
4 0	R:	0.317	0.311	0.311	0.311	0.311	0.311	0.293	0.293	0.293
5 0	I:	0.387	0.585	0.782	0.783	0.783	0.797	0.797	0.804	0.804
5 0	R:	0.160	0.124	0.041	0.041	0.041	0.041	0.041	0.041	0.041
6 0	I:	0.204	0.655	0.674	0.686	0.692	0.692	0.692	0.713	0.713
6 0	R:	0.305	0.066	0.066	0.066	0.066	0.066	0.066	0.066	0.066
7 0	I:	0.953	0.963	0.963	0.963	0.969	0.969	0.969	0.969	0.969
7 0	R:	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006	0.006
8 0	I:	0.558	0.606	0.707	0.707	0.707	0.730	0.736	0.736	0.738
8 0	R:	0.095	0.095	0.095	0.095	0.095	0.089	0.089	0.089	0.089
9 0	I:	0.369	0.451	0.453	0.487	0.502	0.502	0.574	0.582	0.595
9 0	R:	0.160	0.160	0.160	0.130	0.130	0.130	0.130	0.130	0.130

TABLE 16. MUTUAL INFORMATION AND RESIDUES FOR POSTITERATIVE OPTIMIZATION OF MINMAX AGGLOMERATIVE CLUSTERS OF RECOGNITION PATTERNS

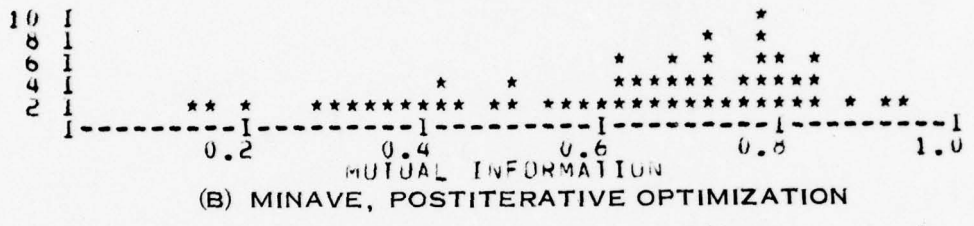
DG/RP	C:	2	3	4	5	6	7	8	9	10
0 0	I:	0.787	0.689	0.678	0.726	0.720	0.764	0.709	0.719	0.734
0 0	R:	0.042	0.060	0.060	0.054	0.060	0.054	0.060	0.066	0.066
1 0	I:	0.310	0.409	0.419	0.362	0.396	0.523	0.596	0.515	0.516
1 0	R:	0.185	0.179	0.167	0.202	0.185	0.149	0.125	0.131	0.137
2 0	I:	0.628	0.466	0.611	0.710	0.690	0.664	0.675	0.795	0.789
2 0	R:	0.083	0.149	0.101	0.065	0.077	0.095	0.083	0.054	0.048
3 0	I:	0.401	0.443	0.473	0.579	0.602	0.602	0.557	0.587	0.620
3 0	R:	0.154	0.136	0.154	0.118	0.107	0.107	0.118	0.124	0.130
4 0	I:	0.153	0.176	0.252	0.239	0.294	0.325	0.370	0.448	0.416
4 0	R:	0.291	0.293	0.234	0.281	0.210	0.234	0.198	0.156	0.174
5 0	I:	0.549	0.735	0.670	0.693	0.757	0.742	0.804	0.805	0.793
5 0	R:	0.095	0.053	0.065	0.077	0.047	0.053	0.041	0.041	0.047
6 0	I:	0.841	0.770	0.764	0.745	0.730	0.736	0.816	0.829	0.812
6 0	R:	0.024	0.048	0.042	0.048	0.054	0.054	0.036	0.048	0.048
7 0	I:	0.953	0.911	0.828	0.887	0.888	0.881	0.869	0.936	0.935
7 0	R:	0.006	0.012	0.030	0.030	0.030	0.024	0.036	0.012	0.012
8 0	I:	0.703	0.797	0.840	0.840	0.800	0.823	0.834	0.828	0.878
8 0	R:	0.054	0.036	0.030	0.030	0.042	0.036	0.042	0.042	0.030
9 0	I:	0.554	0.494	0.505	0.700	0.568	0.519	0.645	0.631	0.704
9 0	R:	0.095	0.148	0.154	0.065	0.107	0.124	0.095	0.124	0.083

TABLE 27. HISTOGRAM OF DIGIT-RECOGNITION PERFORMANCE

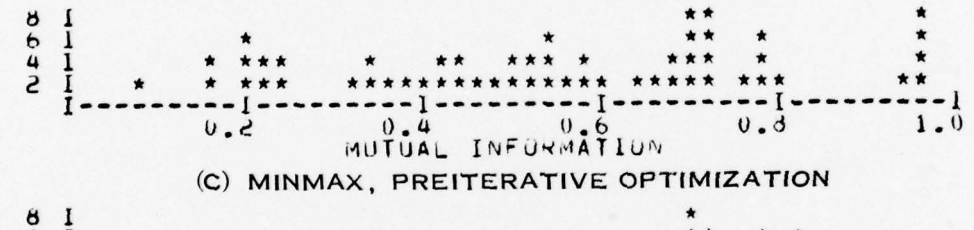
No. of	Percent	Number of	Number of	Number of



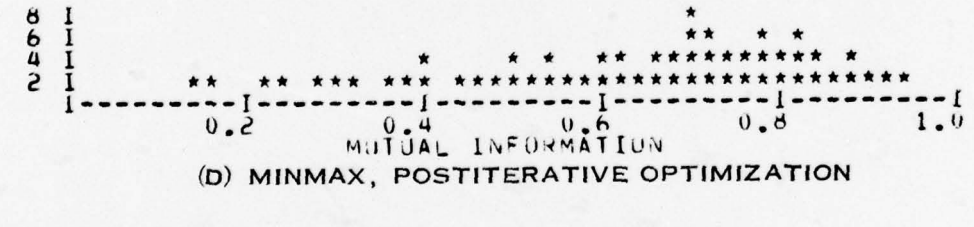
(A) MINAVE, PREITERATIVE OPTIMIZATION



(B) MINAVE, POSTITERATIVE OPTIMIZATION



(C) MINMAX, PREITERATIVE OPTIMIZATION



(D) MINMAX, POSTITERATIVE OPTIMIZATION

Figure 34. Comparison of Mutual *Information for* Clustered Recognition Patterns Using Four Clustering Algorithms

SECTION V

GENERAL-PURPOSE SPEECH I/O CAPABILITY

This section describes the methods used in the AP-120B version of the digit recognition program. Topics covered include the data collection hardware, the filter simulation, the auto-correlation computation, and a discussion of digitizing and playback utilities.

A. SYSTEM DESCRIPTION

The primary impetus of designing this remote speech I/O facility was to relieve the host from the burden of controlling analog-to-digital (A/D) and digital-to-analog (D/A) converters. A second consideration was to develop a method to get data into an array processor at a high rate of speed, to allow real-time data collection and processing. Figure 35 is a block diagram of the resulting speech I/O subsystem.

The configuration consists of a TI 980B host computer, a Floating Point Systems AP-120B array processor, and a TI 990/10 computer with attached A/D and D/A converters. The 990/10 collects and plays out data under the control of a "mailbox" memory location in the AP-120B. Therefore, the 990/10 can be controlled by either the 980B or the AP-120B. The AP-120B is used primarily to reduce the quantity of data by transforming the raw speech to a more compact form (e.g., filtering or preprocessing). In a typical application, the host would request data from the 990/10, request the AP-120B to process the data, and then request that the results of that processing be sent to the 980B.

Software directly used in the I/O subsystem consisted of two parts. The first is the 990/10 software that controls the A/D and D/A, buffers the input and output speech, and controls speech I/O to the AP-120B. The second piece of software runs on the 980B and is basically a device driver for the 990/10. This driver handles the channel protocol as well as all I/O between the 980B and the AP-120B. In addition, existing software that digitizes and edits speech data using the 980B internal A/D and D/A was modified to use the new data acquisition subsystem.

The offloading of the host was carried one step further in the digit-recognition program. In order to free the host from controlling the 990/10 and AP-120B, the AP-120B was put in charge of this entire process. Since the AP-120B has direct memory access (DMA) capability to the

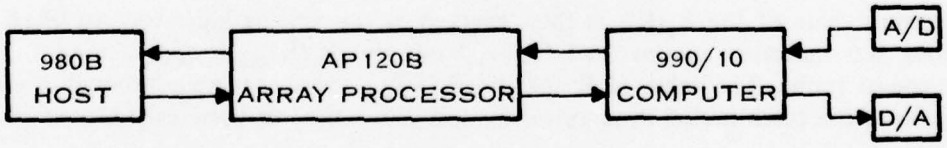


Figure 35. Speech Channel Block Diagram

PRECEDING PAGE NOT FILMED
BLANK

host, the host need only tell the AP-120B where to place the processed data and when to begin collection. This scheme allows continuous input of speech, since host activity is totally disjointed from the data collection or processing. With all this computational burden removed from the host, it can easily keep pace with real-time processing.

B. FILTER SIMULATION IN THE AP-120B

The first process initiated on the input speech data is a recursive filter simulation. The center frequencies and bandwidths of this simulation are designed to match those of the hardware filters specified in Subsection II.A and Appendix A. The filter model included both preemphasis and envelope shaping to match the hardware filters. These 16 filters were typically sampled every 10 milliseconds. This filter simulation accounts for approximately 60 percent of real time when data are collected at an 80-microsecond sample rate.

The output of the filter simulator is then preprocessed, and the output of the preprocessor is sent to the 980B host memory. The preprocessing is the same as that described in Appendix A.

C. VOICING DECISION FROM THE AP-120B AUTOCORRELATION PITCH TRACKER

An estimate of voicing was included in this speech input subsystem by performing an autocorrelation on the input speech. This consisted of sliding a window of speech over previous speech. Given a frame of speech data consisting of N samples, the last M samples of the frame, W_r, are used as a sliding window for a reverse correlation. The normalized cross-correlation, $R_r(K)$, between the sliding window, W_r, and the M speech samples earlier in time starting at the $(M - K_{min})th$ sample is used for this reverse correlation; i.e.,

$$R_r(K) = \frac{\left[\sum_{m=1}^{M} X(M - m + 1)\, X(M - m - K + 1)\right]}{\left(\left[\sum_{m=1}^{M} X^2(M - m + 1)\right]\left[\sum_{m=1}^{M} X^2(M - m - K + 1)\right]\right)} \qquad (28)$$

$$(K_{min} \leq K \leq K_{max})$$

The maximum value of the $R_r(K)s$ is then selected as the voicing indicator. An $|R_r(K)_{max}|$ less than about 0.6 indicates an unvoiced frame, while an $|R_r(K)_{max}|$ approaching 1 indicates a strongly voiced frame. The value of K for the $R_r(K)_{max}$ was not used, although it corresponds to the value of the pitch period in samples. Typical values used in these calculations are:

M = 375 (375 samples at 80 microseconds = 30 milliseconds)

N = 125 (125 samples at 80 microseconds = 10 milliseconds)

K_{min} = 25 (25 samples at 80 microseconds = 2 milliseconds)

K_{max} = 250 (250 samples at 80 microseconds = 20 milliseconds)

SECTION VI
EXPERIMENTAL RESULTS

A. SPEAKER-INDEPENDENT DIGIT RECOGNITION

1. Data Sets

A wide variety of digit-recognition testing was done using two types of data sets. The most heavily used data set was the total-voice evaluation data set. This test data set was part of a data base collected in a sound booth over a 3-month period at Texas Instruments. The test data set was extracted from around the middle of this 3-month period, to avoid the initial microphone fright of the subjects, and consisted of one repetition of one of 10 possible sets of 10 six-digit sequences uttered by 106 subjects (64 males, 42 females). The actual sequences used in this data collection are shown in Table 17. The test data were digitized, edited, and preprocessed during the total voice contract to ensure the precise replicability of the test data. However, since these "test" data were used for multiple experiments to evaluate the effect of parameter variations, the validity of the absolute recognition results is not assured. In addition, the data were also idealized by editing, which avoids spurious false recognition of background noises as true data. For these reasons, further experiments were performed on a second data base.

The second data base used is a subset of a large digit-recognition data base currently being collected in the speech community. The data being collected use sequences devised by Martin and Herscher, modified to include both the "oh" and the "zero" pronunciations of the digit zero. The texts being used in these data collections are shown in Table 18. All the multiple-digit sequences are supposed to be said in a continuous manner, although not all subjects always complied.

2. Digit-Recognition Results for Six-Digit Sequences

A total of 29 evaluation runs were made on the 1,060 six-digit sequences from the total voice evaluation data set. The overall digit recognition rates and conditions for all runs are given in Table 19. Even though no syntactic constraints (except length) were applied during the digit recognition, the total voice evaluation data set used was compatible with the design data since the following digit pairs, all nasal-to-vowel, glide or semivowel (or vice versa) transitions, were disallowed in both data sets:

0–1	1–8	2–9	3–9	4–9	9–1
0–8	2–1	3–1	4–1	7–1	9–8
0–9	2–8	3–8	4–8	7–8	

The first of these evaluation runs (no. 45), was the syntactically unconstrained digit recognition for the final evaluation for the total voice study. A detailed description of the thresholds and parameters for the evaluation runs is given in the total voice final report. The values of most of these parameters remained unchanged during this current study, except as noted in this section. These parameters are listed below along with the values used for run no. 45:

TABLE 17. THE 10 SETS OF 10 SIX-DIGIT SEQUENCES USED IN TESTING

057342	072358	027683	068513	061934
124063	145867	176840	165327	159034
273069	237945	243057	261754	253760
358206	361907	361745	346810	368405
451960	458973	468153	457063	430752
546207	510264	510426	520463	517943
612703	613270	675904	654372	675823
720364	724513	724351	759403	794302
869512	879045	879630	853706	852734
945703	946513	932750	942607	926034
035162	026954	047658	057423	026873
152374	162573	162735	142760	132057
206457	265903	269305	234768	234687
347620	367514	345172	345687	345768
453076	463275	423579	403685	403251
540276	570369	516890	576804	547602
658793	694583	694230	619473	651942
759403	758946	740258	750619	790234
851762	869350	861023	851924	879630
974581	951672	968027	968450	958170

Reference-point location parameters

Peak-to-valley ratio (PVR) = 1.10

Maximum valley point error (Max VPE) = 615

OPTSEQ (valley point sequencing parameters)

dt limits (dt_{max}, dt_{min})
Expected dt ($\hat{dt}$)

see Table 11 of total voice final report[1]

Minimum expected dt (used to determine dt* for the denominator in the point-pair error calculation:

$$dt^* = \max(\hat{dt}, \hat{dt}_{min}) = 4$$

Time deviation weighting (β) = 2

Floor of valley point error (OFFSET) = 100

Hypothesized digit parameters

Minimum absolute average energy across recognition pattern (EN_{min}) = 150

Weighting of sequence error (SQ) contribution to total normalized error for digit k (w_k) = 0.1

TABLE 18. MARTIN-HERSCHER DIGIT TEXTS

Isolated Digits

9	6	2	4	1	7	ZERO	3	5	8	OH
9	6	2	4	1	7	ZERO	3	5	8	OH

Five-Digit Codes
(Pronounce "0" as ZERO.)

08175	10260	55806
67438	44953	32146
29091	60733	68630
91625	81754	79241

(Pronounce "0" as OH.)

08175	10260	55806
29091	60733	68630

Three-Digit Codes
(Pronounce "0" as ZERO.)

(1) 525	(11) 990	(21) 631	(31) 005	(41) 033
(2) 759	(12) 583	(22) 349	(32) 140	(42) 477
(3) 101	(13) 171	(23) 565	(33) 819	(43) 680
(4) 626	(14) 098	(24) 113	(34) 974	(44) 306
(5) 202	(15) 232	(25) 460	(35) 357	(45) 915
(6) 727	(16) 670	(26) 892	(36) 212	(46) 782
(7) 366	(17) 854	(27) 964	(37) 551	(47) 248
(8) 044	(18) 386	(28) 076	(38) 161	(48) 887
(9) 843	(19) 795	(29) 228	(39) 453	(49) 939
(10) 418	(20) 429	(30) 737	(40) 508	(50) 694

(Pronounce "0" as OH.)

101	990	460	005	033
202	098	076	140	680
044	670		508	306

Normalizers to account for expected recognition error for digit k (TE_k)—values given later in this section.

Maximum allowable total normalized error for digit k (NE_k):

Digit	0	1	2	3	4	5	6	7	8	9
$NE_{k_{max}}$	128	97	123	110	116	113	110	109	107	114

TABLE 19. SYNOPSIS OF EVALUATION RESULTS

Run No.	Percent Correct Recognition	Remarks (Changes From Previous Runs)
45	90.5	Baseline (TVBISS final parameters): Multiple reference patterns; minimum energy = 150; 0.1 percent NE thresholds; PVR = 1.1; maximum valley point error = 615; sequence length unconstrained.
46	90.0	Same as run no. 45 except new tree-searching subroutine (DECIDE) used with minimum separation = 3 centiseconds; maximum separation = 80 centiseconds.
47	89.4	Same as run no. 46 except sequence length constrained to 6.
48	89.9	Same as run no. 47 except maximum separation = 120 centiseconds and minimum energy = minimum [150, 0.1 (maximum energy of all hypothesized digits)].
49	90.0	Same as run no. 48 except PVR = 1.05 (for this run only).
50	90.3	Same as run no. 48 except point-pair error between reference points 1 and 3 added.
51	89.3	Same as run no. 50 except minimum energy = minimum [150, 0.2 (maximum energy of all hypothesized digits)].
52	91.8	Same as run no. 50 except total normalized error (NE) for 3 reference-point words multiplied by 0.9 (longer words are, in general, more reliably recognized).
53	91.9	Same as run no. 52 except minimum energy = minimum [150, 0.15 (maximum energy of all hypothesized digits)].
54	92.3	Same as run no. 53 except sequence length unconstrained (for this run only).
55	93.8	Same as run no. 53 except 3-bit quantized T-function added to scanning patterns and maximum valley-point error = 860.
56	93.3	Same as run no. 55 except OFFSET in SQ eliminated (for this run only).
57	94.1	Same as run no. 55 except SQ thresholds = 2,000/1,000.
58	94.1	Same as run no. 55 except maximum valley-point error = 700 and SQ thresholds = (800, 330, 860, 400, 340, 400, 820, 700, 420, 430) for digits 0 through 9, respectively.
59	93.5	Same as run no. 58 except difference data in scanning patterns eliminated.
60	93.4	Same as run no. 59 except middle column of scanning patterns eliminated.
61	94.1	Same as run no. 58 except with minor bug in subroutine DECIDE corrected.
62	94.2	Same as run no. 61 except with minor adjustments to the TE normalizing constants to account for TE changes due to T-function inclusion.
63	94.1	Same as run no. 61 except with minor changes to the quantization levels for T-function (for this run only).
64	91.9	Same as run no. 62 except T-function quantized to 16 levels maximum valley-point error = 1,100; PVR = 1.05; SQ thresholds = 2,000/1,000.

TABLE 19. SYNOPSIS OF EVALUATION RESULTS (Continued)

Run No.	Percent Correct Recognition	Remarks (Changes From Previous Runs)
65	94.9	Same as run no. 62 except TE normalizers modified to account for confusion-matrix entries from run no. 62.
66,67	95.1	Same as run no. 62 except TE normalizers modified to account for confusion-matrix entries from previous run.
68	95.2	
69	95.3	
70-72	95.2	
73	95.3	

The following additional parameters were introduced for the syntactically unconstrained digit sequence recognition algorithm added during the current contract.

Minimum (3-centisecond) and maximum (120 centisecond) interdigit times (times between first reference point of one word and last reference point of previous word)

Minimum acceptable ratio of average recognition pattern energy for each word to maximum average recognition pattern energy for all words = 0.15.

Note from table 19 that poorer results were obtained in experiments that eliminated either the floor (OFFSET) to the valley-point error (run no. 56) or the difference data from the scanning patterns (run no. 59).

The digit recognition results for each digit for selected evaluation results are shown in Table 20. The evaluation run results shown are only those exhibiting significant improvements over previous runs. These improvements occurred for run no. 52 because the total normalized error was lowered for three reference-point words, for run no. 55 because the T-function (see Section III.B for definition) quantized to 3 bits was included in the scanning patterns, and for run no. 73 because the normalizers for the recognition error were modified. (Note from Table 19 that quantizing the T-function to 4 bits in run no. 64 degraded performance.)

The change made for run no. 52 that lowered the total normalized error for longer words was a heuristic justified only by the fact that longer words (those with more reference points) are less likely to be spurious hypotheses. This same philosophy was used in the speaker-dependent recognition. These heuristic normalization constants (HNCs) used were as follows:

No. of reference points	2	3	4	5	6	7
HNC	1.00	0.90	0.81	0.73	0.66	0.59

TABLE 20. DIGIT RECOGNITION RESULTS FOR SELECTED EVALUATION RUNS

	Evaluation Run Number					
Digit	45	50	52	55	62	73
0	91.9	91.9	94.3	94.8	95.4	95.4
1	92.6	94.0	92.9	95.2	93.1	92.4
2	76.7	78.1	88.3	92.7	92.7	94.1
3	89.4	89.3	86.7	89.3	89.2	91.9
4	88.6	88.6	88.2	89.7	90.2	97.2
5	98.5	96.2	96.1	97.6	98.1	95.9
6	95.8	95.2	97.8	98.3	98.3	98.0
7	83.7	84.4	89.9	93.8	94.4	98.0
8	97.5	96.8	93.3	97.3	98.5	93.8
9	89.8	89.1	88.0	89.3	91.2	93.7
Overall	90.5	90.3	91.8	93.8	94.2	95.3

The inclusion of the T-function in the scanning pattern was prompted by vowel/nasal reference points being moved into the nasal rather than being at the phoneme boundary. This was primarily caused by the inclusion of reference patterns to accommodate nasalized vowels. Oftentimes, even words having non-nasalized vowel-to-nasal transitions produced lower valley-point errors matching a portion of the nasalized vowel-to-nasal reference pattern. Since the desire was to favor choosing reference-point candidates at the locations of T-function peaks in the input, such a bias could be provided by including an inversely quantized value of the T-function in conveniently unused 4-bit fields in the scanning pattern for the input (Figure 36). Since the corresponding 4-bit fields of the reference scanning patterns were zero, the inverse quantization of the T-function meant that the larger T-function values (lower inversely quantized values) that usually occur at phoneme boundries would produce lower scanning errors relative to those produced when the spectral or energy change was not so great during more nearly steady-state portions of the word. The quantization thresholds given in Table 21 were derived from a cumulative distribution plot of T-function values at the selected reference points in the digit recognition design data and at ±1 and ±2 time samples around those points.

In addition to the improved recognition performance shown for run no. 55 in Table 20, the benefit of including the T-function is the scanning pattern can also be seen by the decrease in the average recognition pattern error, indicating improved time registration of the input speech. This decrease is shown in Table 22.

The third performance improvement was prompted by the confusion matrix for evaluation run no. 62 (Table 23). Four quite large nonsymmetrical substitutions are shown in the off-diagonal entries of the confusion matrix. Since the digits are selected that minimize the minimum total normalized error across the sequence, adjustment of the relative errors among digits will affect the distribution of substitutions in the confusion matrix. The mechanism for performing this adjustment can be seen from the following equation for the total normalized error for the digit k:

$$NE_k = HNC_k \left[\frac{TE_k/\text{no. of column in digit k}}{TE_k \text{ normalizing constant}} \right] + \frac{w_k}{NPP} \sum_{i=1}^{NPP} PPE_i \qquad (29)$$

where HNC is the heuristic normalizing constant, TE is the recognition pattern error, NPP is the number of reference-point pairs, PPE is the point-pair error between two reference points, and w is a weighting constant for the sum of the PPEs.

The TE_k normalizing constants are calculated from the expected values of TE_k as follows:

$$TE_k \text{ normalizing constant} = \frac{\hat{TE}_k/\text{no. columns in k}}{\frac{1}{10}\sum_{i=0}^{9}(\hat{TE}_i/\text{no. columns in i})} \qquad (30)$$

Five sets of values for these normalizing constants are given in Table 24, derived from five different sources:

(1) $E_e + E_s + E_a$ from Table 20 of the Speaker Verification III report

(2) Values of J_e for the number of reference patterns chosen in the total voice study for each digit

(3) Values of TE for each of the digits in correct sequences in the 6-digit sequence evaluation data set for run no. 33

(4) Same as source 3, for run no. 57, which includes the T-function in the scanning patterns

(5) Values derived from incrementally changing the values from source 4 during run nos. 62 through 73.

Although the normalizing constants derived from source 5 will certainly give somewhat biased results since they are tuned to the evaluation set, it should be remembered that the test set is reasonably large (106 speakers). An independent test on the second set of data described in Subsection VI.A.1 showed that, while not achieving the 19-percent reduction in error rate on the six-digit sequences between run nos. 62 and 73, a 6-percent reduction in error rate was achieved using the normalizing constants from source 5 from that achieved using those from source 3.

The confusion matrix for the final evaluation run on this study (no. 73) is shown in Table 25.

One final observation made on the results of the evaluation runs was the usual problem of poorer performance for females than for males, as shown both by the overall recognition results in Table 26 and by the histogram of digit recognition performance (table 27), both from evaluation run no. 73.

3. Digit-Recognition Results for Three-Digit Sequences

Although the data used for the testing reported in the last subsection were from a large number of subjects of different ages, races, dialects, and educational backgrounds, the multiple

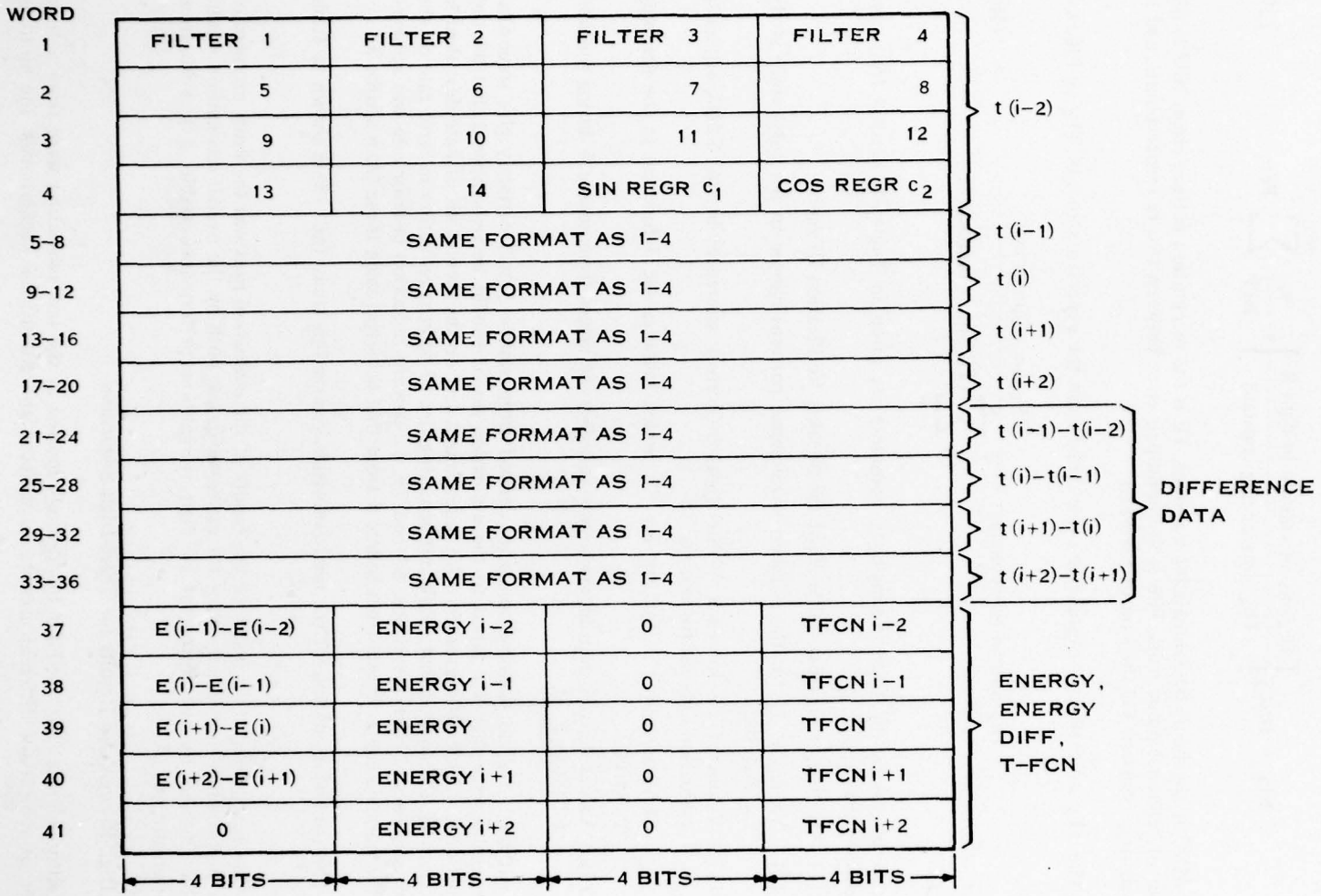


Figure 36. Scanning Pattern Format

TABLE 21. T-FUNCTION QUANTIZATION THRESHOLDS

Quantized Value	Range of T-Function
0	214–∞
1	163–213
2	128–162
3	101–127
4	79–100
5	60–78
6	40–59
7	0–39

TABLE 22. DECREASE IN AVERAGE RECOGNITION PATTERN ERROR BY *INCLUDING* T-FUNCTION IN SCANNING PATTERNS AVERAGE RECOGNITION PATTERN ERROR

Digit	Run No. 45 (No T-Function)	Run No. 57 (3-Bit T-Function)
0	359.8	354.0
1	440.4	409.5
2	316.9	311.3
3	294.3	289.0
4	221.5	218.8
5	431.4	424.9
6	283.3	275.2
7	423.4	415.4
8	287.3	288.3
9	444.7	435.3

TABLE 23. CONFUSION MATRIX FOR DIGIT RECOGNITION FOR 6-DIGIT SEQUENCES FOR RUN NO. 62

		Recognized										
		0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**X**
Said	0	660	8	1	10	1	2	3	—	1	3	—
	1	4	403	1	12	6	3	—	1	2	2	—
	2	2	1	581	21	1	—	3	—	[17]	—	—
	3	8	1	21	657	4	—	—	—	[46]	1	—
	4	1	13	—	4	644	[49]	—	—	—	—	—
	5	1	3	1	1	2	779	—	—	1	—	8
	6	2	—	—	4	—	—	752	—	5	—	1
	7	3	7	2	2	—	4	—	719	[18]	1	—
	8	—	—	—	3	1	—	2	—	398	—	1
	9	2	9	1	12	1	2	—	—	11	402	1
	X	—	1	—	1	—	—	—	—	2	—	—

TABLE 24. RECOGNITION ERROR (TE) NORMALIZING CONSTANTS

Digit	Source 1	Source 2	Source 3	Source 4	Source 5
0	1.039	1.022	0.998	1.005	1.020
1	1.170	1.089	1.225	1.163	1.150
2	0.708	0.837	0.878	0.884	0.890
3	1.085	1.016	1.021	1.026	1.020
4	0.853	0.719	0.681	0.691	0.738
5	1.182	1.292	1.195	1.207	1.010
6	0.705	0.870	0.788	0.782	0.730
7	1.052	0.945	0.977	0.983	1.170
8	1.007	1.133	1.008	1.024	0.860
9	1.200	1.076	1.229	1.236	1.330

TABLE 25. CONFUSION MATRIX FOR DIGIT RECOGNITION FOR 6-DIGIT SEQUENCES FOR RUN NO. 73

		Recognized										
		0	1	2	3	4	5	6	7	8	9	X
Said	0	657	4	1	11	2	1	3	1	---	3	1
	1	4	401	---	11	12	---	---	1	---	4	1
	2	2	1	589	21	1	---	2	2	7	1	---
	3	11	1	23	678	8	---	---	1	14	1	---
	4	---	9	---	1	691	9	---	1	---	---	---
	5	---	3	---	---	8	763	---	---	---	1	21
	6	2	---	1	7	2	---	749	1	1	---	1
	7	2	2	1	1	1	1	---	741	4	1	1
	8	---	1	---	4	2	---	7	6	380	1	4
	9	5	4	1	10	---	1	---	---	5	413	2
	X	---	1	1	1	---	---	---	---	---	2	---

TABLE 26. DIGIT-RECOGNITION PERFORMANCE OF MALES AND FEMALES

	Percent Correct Recognition	
Digit	Males	Females
0	97.8	91.6
1	92.8	91.8
2	94.4	93.6
3	94.0	88.2
4	99.8	93.1
5	97.0	94.1
6	99.1	96.7
7	99.2	96.1
8	94.9	92.4
9	95.5	91.4
Average	96.7	93.1

TABLE 27. HISTOGRAM OF DIGIT-RECOGNITION PERFORMANCE

No. of Errors	Percent Correct	Number of Subjects	Number of Males	Number of Females
0	100	22	19	3
1	98	20	14	6
2	97	21	13	8
3	95	11	6	5
4	93	11	7	4
5	92	4	1	3
6	90	5	0	5
7	88	4	2	2
8	87	3	1	2
9	85	0	0	0
10	83	3	0	3
11	82	0	0	0
12	80	0	0	0
13	78	1	1	0
14	77	0	0	0
15	75	1	0	1
≥ 16	<75	0	0	0

evaluations using these same data made a further independent test mandatory. Results in this section use the three-digit sequences of the second data base, excluding those with the "oh" pronunciations. Results in the next subsection are for the digits said in isolation from the same data base.

Note that, for these 50 sequences, all digits appear an equal number of times and in all contexts of preceding and following digits. Since the original application of the work done in the total voice study was for syntactically constrained sequences, not all digit pairs were used in the design data, as described in the previous subsection. Hence, the recognition performance is expected to be poorer for digits involved in these transitions. This poorer performance does not, however, reflect on the method developed for choosing reference patterns, but only on the inadequacy of the design data for unconstrained digit recognition.

The data used in these tests were collected in sound booths or sound-treated rooms at two locations: Texas Instruments (Dallas, Texas) and the Institute for Advanced Study of the Communication Process (Gainesville, Florida).

Since the poorer performance of females has been demonstrated in the previous subsection (as well as in all other word-recognition studies that have been done, to the best of the author's knowledge), experiments were performed for male subjects only, 12 from Dallas and 11 from Gainesville. The recognition performance is shown as the far right column in the confusion matrix in Table 28. The overall percent correct is 94.0.

It has been noted in these studies that the confusion matrix can be quite speaker-dependent. Since, in the 3-digit sequence data base, there were more digits (150 versus 60) and fewer subjects (23 versus 106), the confusion matrix entries are more susceptible to high substitution rates by particular speakers. For example, 23 of the 48 3-for-2 substitutions shown in Table 28 were caused by two speakers. However, the two digits with the greatest reduction in recognition rate (2 and 8) from that given in Table 26 for males are two of the four digits

TABLE 28. CONFUSION MATRIX FOR DIGIT RECOGNITION FOR 3-DIGIT SEQUENCES CONSTRAINED IN LENGTH . .

Said	Recognized 0	1	2	3	4	5	6	7	8	9	X	Percent Correct
0	340	1	---	---	---	---	---	---	---	3	---	98.8
1	1	331	---	2	3	---	---	---	---	8	---	95.9
2	7	1	287	48	---	---	---	---	---	1	1	83.2
3	3	3	5	327	---	---	---	---	1	6	---	94.8
4	1	5	---	1	330	6	---	---	---	---	1	95.9
5	---	---	---	---	---	341	---	---	---	1	3	98.8
6	14	---	1	1	1	---	323	3	1	---	---	94.0
7	---	1	1	1	---	---	---	341	1	---	---	98.8
8	---	2	1	29	1	---	5	2	294	10	1	85.2
9	---	11	---	---	1	---	---	---	6	326	1	94.5
X	---	---	---	1	---	---	1	---	---	2	---	---

having reference points located at the word boundaries. Hence, if the reference scanning patterns used for locating these points do not explicitly account for all allowable contexts, then these reference points may be missed during scanning because of the lack of a significantly deep valley in the scanning error (distance to the reference scanning pattern). In such a case, even though the time-normalized recognition pattern is much less affected by context, this digit would not even be hypothesized because of the missing reference point.

Since the percent correct achieved for the 3-digit connected digits was 94.0 using reference patterns that did not account for all transitions, it is reasonable to assume that the 96.7-percent correct achieved for males in the 6-digit sequences was minimally, if any at all, caused by tuning to the test set during the parameter evaluations done on the 6-digit sequence test set. It appears reasonable that such a recognition rate could be achieved on the unconstrained digit recognition if the reference scanning pattern set were expanded to include patterns to account for these transitions. Such patterns could be generated with an expanded design set using the clustering techniques developed in these studies.

4. Digit Recognition Results for Isolated Digits

A very limited experiment was run to test the digit recognition on isolated digits. The test involved two samples of each of the 10 digits said in isolation from the same 23 speakers used in the 3-digit sequence test. In view of the recognition rates achieved on continuous speech, the recognition rate of the isolated digits was a surprisingly low 95.4 percent, with over one-third of the errors being caused by 3-for-2 and 5-for-4 substitutions. More expected would be results such as that achieved by Martin[2], whose error rate was approximately halved between his test T-2 (Philadelphia–connected digits) and his test T-3 (Philadelphia–isolated digits).

One possible explanation for the higher-than-expected isolated digit error rate is that the patterns contained in the reference set to account for contextual variations are generating spurious hypotheses during isolated word recognition.

The specific confusion matrix for the isolated digits is given in Table 29.

TABLE 29. CONFUSION MATRIX FOR DIGIT RECOGNITION OF ISOLATED DIGITS

Said	Recognized 0	1	2	3	4	5	6	7	8	9	Percent Correct
0	45	—	—	1	—	—	—	—	—	—	97.8
1	—	45	—	—	—	—	—	—	—	1	97.8
2	—	—	42	4	—	—	—	—	—	—	91.3
3	—	—	—	46	—	—	—	—	—	—	100.0
4	—	2	—	—	40	4	—	—	—	—	87.0
5	—	—	—	—	—	46	—	—	—	—	100.0
6	—	—	—	1	—	—	45	—	—	—	97.8
7	—	—	—	—	—	—	—	44	—	2	95.7
8	—	—	—	3	—	—	—	—	43	—	93.5
9	—	—	—	—	—	1	—	—	2	43	93.5

5. Effect of Spectral Normalization Technique on Digit-Recognition Performance

Subsequent to the performance tests described previously in this section, investigation of the high 3-for-2 substitution rate in the 3-digit sequences revealed a mechanism for improving recognition results and for making them less susceptible to variations in background noise. This investigation revealed that the valley point error for reference point 1 for the digit 2 (i.e., the silence/plosive transition) had a higher error for the real-time data than for the digitized data. This was found to be caused, at least in part, by different "silence" spectra. The differences were alleviated to some extent by increasing σ_{min} in the constant σ_j^* used in normalizing the regressed filter outputs (see discussion in Appendix A):

$$\sigma_j^* = \sigma_{post_j} + \sigma_{min} \tag{31}$$

where σ_{post_j} is the post-regression standard deviation.

The specific effects on the spectrum of changing σ_{min} can be seen from the three spectra shown in Figure 37 for the word two as said by J.S. The increase in the number of hypothesized digits and the decrease in the total normalized errors for the six digits in the sequences from which the spectra in Figure 37 were extracted are shown in Table 30. A further breakdown into the terms that make up the total normalized errors is given in Table 31. Since these tests were done directly from the analog tape for each trial, exactly repeatable filter outputs are not obtained. However, general trends can be noted from these tables, such as the over 50-percent drop in the valley point error for reference point 1 of the digits 5, 2, and 4, which contain silence or low energy frication (/f/). An additional point of interest is the consistently lower recognition errors for the digitized data than for the analog data.

The expected benefits to be derived from using a larger σ_{min} are twofold. First, the normalized spectrum will tend to be more even for silence or low-energy fricatives, making the resulting patterns more resistant to variations in background noise. Second, since the reference

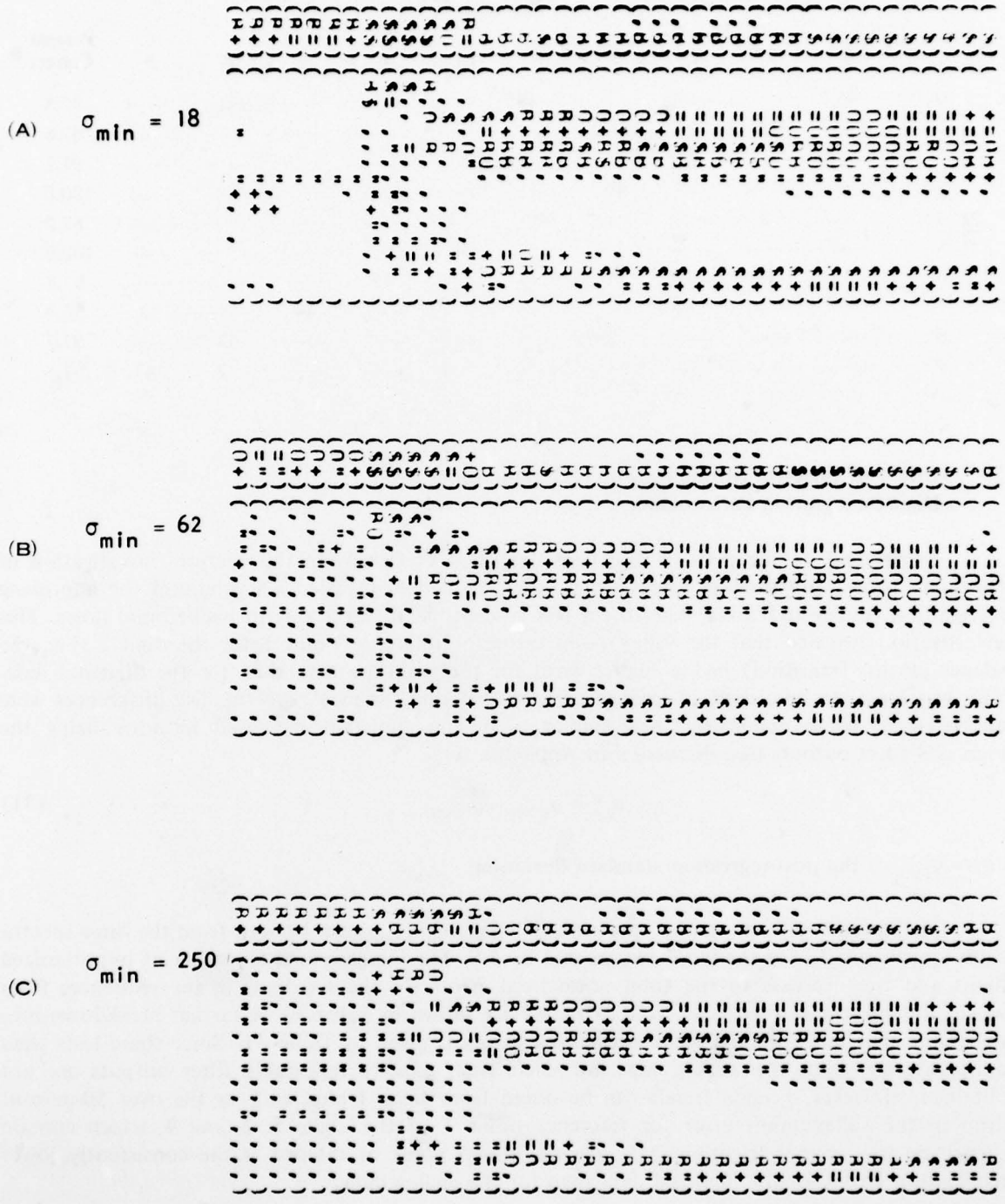


Figure 37. Spectra for Digit "Two" for Speaker J.S.

TABLE 30. TOTAL NORMALIZED ERROR (NE) FOR DIGITS FROM SEQUENCE 852–734 FOR SPEAKER J.S.

σ_{min}	No of Hypothesized Digits	Normalized Error 8	5	2*	7*	3	4
18	49	55	55	63(70)	48(54)	59	66
37	58	60	49	49(55)	47(53)	64	57
62	52	54	44	45(51)	50(56)	52	55
100	58	51	41	45(50)	46(52)	58	51
150	76	51	38	40(45)	48(54)	52	46
200	70	40	39	42(47)	45(51)	50	45
250	84	38	38	41(46)	45(51)	47	45
375	72	40	33	36(41)	45(50)	43	44
Run No. 53 (σ_{min} = 62)	~52	47	38	40(45)	44(49)	46	54

*Since three reference point digits are multiplied by 0.9, the unadjusted errors are given in parentheses.

TABLE 31. VALLEY POINT ERRORS, SEQUENCE ERRORS (SQ), AND RECOGNITION ERRORS (TE) FOR DIGITS FROM SEQUENCE 852–734 FOR SPEAKER J.S.

	8				5				2				
σ_{min}	1	2	SQ	TE	1	2	SQ	TE	1	2	3	SQ	TE
18	389	161	256	335	385	225	312	468	442	330	320	657	418
37	346	149	216	395	265	218	252	430	308	253	293	474	340
62	338	137	202	358	264	200	214	389	291	261	268	447	320
100	250	124	152	346	279	167	198	373	255	287	250	404	328
150	324	130	190	331	181	188	158	359	206	244	231	326	297
200	240	105	136	268	214	165	164	368	213	271	231	367	308
250	210	111	126	260	218	163	166	362	181	257	273	358	298
375	212	103	122	275	191	120	124	318	195	247	220	313	273
Run No. 53	303	143	190	309	222	219	200	343	144	261	253	305	308

	7					3				4			
σ_{min}	1	2	3	SQ	TE	1	2	SQ	TE	1	2	SQ	TE
18	280	272	148	353	489	406	352	448	306	376	129	214	336
37	336	267	153	392	468	420	278	404	358	239	129	152	302
62	352	263	220	487	471	374	320	388	266	193	114	122	298
100	318	255	143	382	462	350	271	356	329	202	108	122	278
150	300	259	125	359	488	314	251	310	296	181	88	102	251
200	291	247	122	324	464	306	238	294	290	189	92	108	248
250	238	252	117	289	478	267	222	252	276	179	97	108	244
375	274	239	108	287	475	273	200	230	257	181	101	110	231
Run No. 53	322	306	160	418	432	307	241	284	267	304	133	182	280

patterns used in speaker-independent digit recognition result from averaging patterns from many speakers, the reference patterns appear more "washed-out," lacking the sharp contrasts found in speaker-specific patterns. Hence, a flatter spectrum on the input speech resulting from using a larger σ_{min} would probably result in a better match to speaker-independent reference patterns.

This hypothesis was tested using a tape generated at RADC (in the computer room containing speech-processing equipment) consisting of two repetitions each of two speakers (R.V. and J.F.) of the 50 three-digit sequences given in Table 18. The recognition results for all four repetitions are given in Table 32 using both a σ_{min} of 62 and a σ_{min} of 250. The recognition results for the larger σ_{min} show a small improvement. A performance improvement would correspondingly be expected on the results presented in the previous subsections since all these results used data preprocessed using a σ_{min} of 62.

TABLE 32. σ_{min} VARIATION PERFORMANCE TEST

		Length Constrained		Length Unconstrained	
Subject	**Session**	**σ_{min} = 62**	**σ_{min} = 250**	**σ_{min} = 62**	**σ_{min} = 250**
J.F.	1	96.0	93.3	98.0	95.3
J.F.	2	95.3	97.3	96.7	96.7
R.V.	1	89.3	90.0	90.0	92.0
R.V.	2	91.3	94.0	92.0	92.0
Overall		92.5	93.7	94.2	94.0

In conclusion, an observation made concerning sequence recognition during the testing of these two subjects should be noted. In an operational system, where sequences can be repeated, the only consequence of rejected sequences is a decrease in throughput (assuming a sequence can finally be accepted if repeated); therefore, rejections should not be used in calculation of the percent correct recognition rate. The percent correct sequence recognition rate is then given by

$$\% \text{ correct} = \frac{\text{no. of correct sequences}}{\text{no. of correct sequences + no. of incorrect sequences}}$$

or

$$\% \text{ correct} = \frac{\text{no. of correct sequences}}{\text{no. of sequences uttered} - \text{no. of rejected sequences}}$$

In the case where the length is constrained, the present tree-searching algorithm described in Section II chooses the best sequence of the specified length. However, the sequence recognition results for the limited testing given in this subsection indicate that by modifying the algorithm so that the sequence is accepted only if the length of the best sequence is the same as the specified length, the sequence recognition rate (as defined above) would improve as shown in the following table:

AD-A074 206 TEXAS INSTRUMENTS INC DALLAS F/G 17/2

LIMITED VOCABULARY CONTINUOUS WORD RECOGNITION.(U)

AUG 79 R L DAVIS, B G SECREST, C J CATO F30602-77-C-0139

UNCLASSIFIED RADC-TR-79-204 NL

2 OF 2

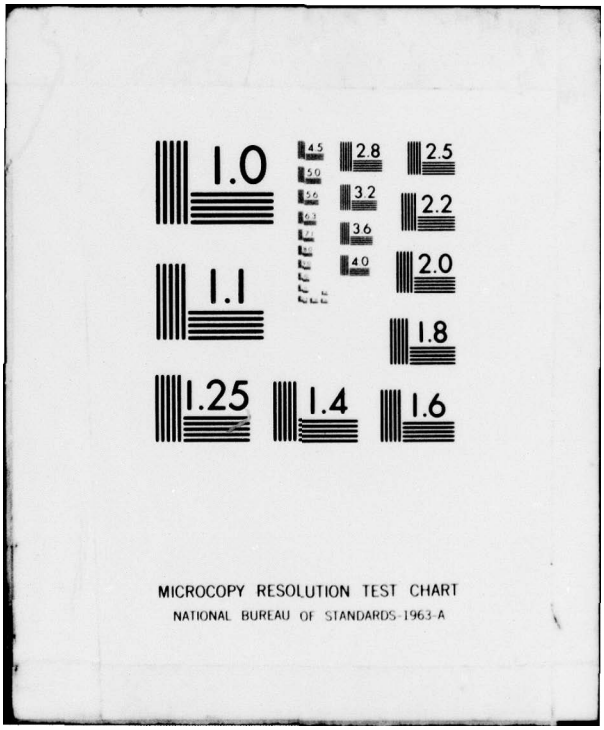
AD A074206

END

DATE FILMED

10-79

DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

No. of Utterances	Best Sequence of Length 3	If Best Sequence Is of Length 3
Correct	336	330
Rejected	7	25
Incorrect	57	45
Percent correct	85.5	88.0

B. LIMITED VOCABULARY WORD-RECOGNITION EXPERIMENT

Five speakers from the Speech Research Branch at Texas Instruments were tested on the limited word-recognition algorithm using an automatic enrollment and a hand enrollment. Each speaker was recorded onto analog tape while seated in the sound booth. An enrollment session collected from each speaker consisted of four discrete repetitions of the following words:

Zero	Five	Minus	Hundred
One	Six	Plus	Thousand
Two	Seven	Point	Enter
Three	Eight	Backup	Erase
Four	Nine	Punch	Display

At some time later the same day or the next day, an execution session was collected from each speaker. The execution session consisted of a set of 20 phrases of three randomly chosen words and a set of 20 phrases of random lengths (up to seven words) of randomly chosen words. Each phrase in the execution session was spoken continuously. Two of the speakers had two execution sessions spaced a half day apart.

The speakers were then enrolled off-line using both automatic enrollment and hand enrollment. The execution sessions were then tested against these enrollments. The results of the experiment are given by the confusion matrices of Tables 33 and 34. The left of the matrix shows what was said and the top of the matrix shows what was recognized. An entry in the "X" column means nothing was recognized (a deletion). The entries in the matrix are the number of times a word was recognized versus what was said. A compilation of the results for each inidividual speaker is given in Table 35. One of the speakers (Keith) had two execution sessions, and his first execution was used for a supervised updating. The second execution session was used against the updated reference patterns. The results are given in the last column of Table 35.

TABLE 33. CONFUSION MATRIX FOR AUTOMATIC ENROLLMENT

Said \ Recognized	0	1	2	3	4	5	6	7	8	9	–	+	·	BU	PU	HU	TH	EN	ER	DIS	X
0	36	—	—	—	—	—	2	—	—	—	—	—	—	—	—	—	—	—	—	—	4
1	—	29	—	—	1	—	—	—	—	—	—	6	2	—	—	—	—	—	—	—	2
2	—	—	34	2	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	28
3	—	—	2	32	—	—	—	—	2	—	1	1	2	—	—	—	—	—	1	—	9
4	—	—	—	—	32	—	—	—	—	—	—	1	—	—	—	—	—	1	—	—	11
5	—	—	—	—	—	23	—	—	—	—	1	3	—	—	—	—	—	—	—	—	12
6	—	—	—	—	—	—	43	—	—	—	—	—	—	—	—	—	—	—	—	—	4
7	—	—	—	—	—	—	2	44	—	—	—	2	—	—	—	—	—	—	—	—	5
8	—	—	—	—	—	—	1	—	34	—	—	—	—	—	—	—	—	—	—	—	2
9	—	—	—	—	—	—	—	—	—	45	1	—	—	—	—	—	—	—	—	—	8
–	—	—	—	—	—	—	—	1	—	—	38	—	—	—	—	—	—	—	—	—	3
+	—	—	—	—	—	—	—	—	—	—	—	45	—	—	—	—	—	—	—	—	2
.	—	—	—	—	—	—	1	—	—	—	—	—	30	1	—	—	—	—	1	—	7
BU	—	—	—	—	—	—	—	—	—	—	—	—	—	51	—	—	—	—	—	—	2
PU	—	2	—	—	—	—	—	—	—	—	—	1	—	—	25	1	—	—	—	—	11
HU	—	—	—	1	—	—	2	—	—	—	—	—	1	—	—	42	—	—	—	—	7
TH	—	—	1	—	—	—	—	—	—	—	—	1	—	—	—	—	31	—	—	—	1
EN	—	—	—	—	—	—	1	—	—	—	—	—	—	—	—	2	—	33	—	—	11
ER	—	—	—	1	—	—	—	—	2	—	—	—	—	—	—	1	—	—	48	—	1
DIS	—	—	—	—	—	—	—	—	—	—	—	—	1	—	—	—	—	—	—	44	1

TABLE 34. CONFUSION MATRIX FOR AUTOMATIC ENROLLMENT

Said \ Recognized	0	1	2	3	4	5	6	7	8	9	–	+	·	BU	PU	HU	TH	EN	ER	DIS	X
0	35	—	—	1	—	—	—	—	—	—	—	—	—	—	—	—	—	1	—	—	3
1	—	36	—	—	—	—	—	—	—	—	—	—	4	—	—	—	—	—	—	—	2
2	—	—	41	1	—	—	—	—	3	—	—	—	—	—	—	—	—	4	—	—	16
3	—	—	1	34	—	—	—	—	2	—	—	—	—	—	—	—	—	1	—	—	10
4	—	—	—	—	33	—	—	—	—	—	—	1	1	—	—	—	—	—	—	—	12
5	—	—	—	—	—	39	—	—	1	—	—	—	—	—	—	—	—	—	—	—	2
6	—	—	—	—	—	—	43	—	1	—	—	—	—	—	—	—	—	1	—	—	3
7	—	—	—	—	—	—	—	56	—	—	—	—	—	—	—	—	—	—	—	—	1
8	—	—	—	—	—	—	—	—	37	—	—	—	—	—	—	—	—	—	—	—	2
9	—	1	—	1	—	—	—	—	—	51	—	—	—	—	—	—	—	—	—	—	—
–	—	2	—	—	—	—	—	—	—	—	39	—	—	—	—	—	—	—	—	—	5
+	—	—	—	—	—	—	—	—	—	—	—	41	—	—	—	—	—	—	—	—	4
·	—	1	—	—	1	—	—	—	—	—	—	—	37	—	—	—	—	—	—	—	1
BU	—	—	—	—	—	—	—	—	—	—	—	—	—	52	—	—	—	—	—	—	—
PU	—	1	—	—	—	—	—	—	—	—	—	—	3	—	34	—	—	—	—	—	3
HU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	50	—	—	—	—	4
TH	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	34	—	—	—	7
EN	—	—	—	1	—	—	—	—	—	—	1	—	—	—	—	—	—	29	—	—	12
ER	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	53	—	—
DIS	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	46	—

TABLE 35. PERCENT CORRECT SPEAKER-DEPENDENT RECOGNITION RESULTS FOR CONTINUOUS UTTERANCES FROM A 21-WORD VOCABULARY ENROLLED ON ISOLATED WORDS

Speaker	Automatic Enrollment	Hand Enrollment	Updating (Hand)
Gene	77	78	
Richard	73	88	
Louise	74	87	
George	76	94	
Keith	66	80	90

SECTION VII

CONCLUSIONS AND RECOMMENDATIONS

The three major areas of research during this study contract were

(1) High-performance, speaker-independent, connected-digit recognition for syntactically unconstrained digit sequences

(2) Clustering algorithms for use in the development of sets of reference patterns for speaker-independent word recognition

(3) Automatic enrollment for speaker-dependent, connected-word recognition for syntactically unconstrained word sequences.

The program culminated in the installation of the speaker-independent, connected-digit recognition program on the BISS-ADM speaker verification system at RADC using the total voice reference patterns for compatibility. In addition, a long-standing hardware failure with the digital filters on the BISS-ADM system was corrected, resolving performance discrepancies between the systems at RADC and Texas Instruments.

As part of the three tasks, several developments resulted that are generally applicable to the speech technology used in this study. The first of these is a modification to the algorithm for searching the table of hypothesized words (directed graph) that significantly reduces the processing time. The second development is a technique (transparent to previous programs) for including a measure of the spectral transitionitivity (T-function) in the scanning patterns for the purpose of improving the time registration of reference-point locations. The third development is the capability of digitizing and playing back speech data through A/D and D/A connections to the fast array processor. This provides the basis for the fourth development, which is simulation of the digital filters in the array processor, allowing parametric variation of the filter-bank definition and the consequent ability to perform a variety of tests with data that can be more precisely replicated using a variety of filter-bank definitions. The new speech channel capability was also necessary for a fifth development, that of using a quantized autocorrelation value out of an autocorrelation pitch tracker previously implemented on the array processor to produce a "soft" voicing decision for each frame of filtered speech data for eventual incorporation into the time-normalized recognition pattern. The sixth general development came as a natural extension to the capabilities provided by the speech channel and filter simulation. This development is the programming of the preprocessing function in the array processor and subsequent amalgamation of digitizing, filtering, and preprocessing in the array processor for inputting preprocessed speech data to the word-recognition programs. This capability reduces the 980B processing time by about 35 percent, allowing the word recognition algorithm that uses the new directed-graph searching algorithm to operate sufficiently fast to allow continuous speech input without having to discontinue sampling after the input of an utterance.

The speaker-independent, connected-digit recognition portion of this study resulted in a significantly faster algorithm with a 50-percent decrease in error rate over the course of this study—from 90.5 percent correct recognition to 95.3 percent on an evaluation data set of ten 6-digit sequences from 106 speakers (64 males, 42 females).

The development of the clustering algorithm resulted in a two-stage, four-path algorithm with the mechanisms for detecting outlying data points in the design data and with subsequent analysis routines for comparing the results from the various paths and testing the validity of

resulting clusters on the basis of comparisons with *a priori* information about the design data set. The results of the analysis of the digit-recognition design data set revealed that, although the clusters selected during the total voice speaker verification contract generally were good partitions of the data, use of partitions resulting from other paths in the more comprehensive algorithm would have resulted in somewhat more compact clusters in terms of minimizing the sum-of-squared error.

The research into development of an automatic enrollment technique for speaker-dependent word recognition resulted in a method that yielded very good results for isolated word recognition but less acceptable results when used in continuous speech from the same speaker. The better results achieved with comparable hand enrollments point to the desirability of a semiautomated enrollment procedure allowing the operator the option of modifying reference-point locations and recognition-pattern format definitions defined by an automated front end. Independent of the enrollment method, however, the benefit of reference file updating as a means of accommodating contextual variability, as well as intersession variability, became abundantly clear.

Throughout all three phases of this study, the general limitation existed of an insufficient speech data sample rate and spectral resolution of the filter bank, especially in the higher frequency bands. This limitation must be removed before any further word recognition development. In addition, although all recognition features up to this point have been spectral amplitudes or direct correlates thereof (regression coefficients and energy), it is time that more features are used. This, in fact, was the impetus behind the addition of the "soft" voicing decision (quantized autocorrelation coefficient) to the spectral parameters derived during preprocessing.

Care must be taken, however, that none of the new features added are subject to measurement errors sufficient to actually degrade performance. In addition to not degrading *overall* performance, new features must not degrade performance of the poor speakers while improving the results for the good speakers.

However, new features such as autocorrelation values or formant values will require computation capabilities exceeding those of a 16-bit minicomputer. The recommendation for future word-recognition development is that such research be done with a computing facility that

(1) Is capable of fast arithmetic both for longer word-length integers and for floating-point numbers

(2) Contains a large (¼ to ½ million words) primary storage with virtual memory capability

(3) Contains an operating system with more programmer-directed features than typically available on 16-bit minicomputers, allowing more of the time now spent on program development to be spent on speech algorithm development

(4) Contains a fast array processor capable of performing the filter simulation, linear predictive coefficient (LPC) computations, formant tracking, autocorrelation computation, etc., necessary for the extended feature set that is required for further recognition performance improvement.

REFERENCES

1. R.L. Davis, B.M. Hydrick, and G.R. Doddington, "Total Voice Speaker Verification," Rome Air Development Center Technical Report, RADC-TR-78-260, January 1978, A065160.

2. T.B. Martin, "Acoustic Recognition of a Limited Vocabulary in Continuous Speech," Ph.D. Dissertation, University of Pennsylvania, 1970.

3. D.R. Reddy, "Speech Recognition by Machine: A Review," *Proceedings of the IEEE*, 64:501–531, April 1976.

4. G.R. Doddington, "A Method of Speaker Verification," Ph.D. Dissertation (Thesis), University of Wisconsin, 1970.

5. H. Sakoe and S. Chiba, "A Dynamic Programming Approach to Continuous Speech Recognition," *Proceedings of the 7th International Congress on Acoustics*, August 1971.

6. V.M. Velichko and N.G. Zagoruiko, "Automatic Recognition of 200 Words," *International Journal Man-Machine Studies*, 2:223, June 1970.

7. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-23:67–72, February 1975.

8. G.M. White and R.B. Neely, "Speech Recognition Experiments With Linear Prediction, Bandpass Filtering, and Dynamic Programming," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-24:183–188, April 1976.

9. H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-26:43–49, February 1978.

10. B.T. Lowrre, "The HARPY Speech Recognition System," Ph.D. Dissertation (Thesis), Carnegie-Mellon University, 1976.

11. G.M. White, "Continuous Speech Recognition: Dynamic Programming, Knowledge Nets and HARPY," Paper 28-2, *1978 WESCON Professional Program*, September 1978.

12. J.E. Porter, "LISTEN: A System for Recognizing Connected Speech Over Small, Fixed Vocabularies, in Real Time," Naval Training Equipment Center Technical Report, NAVTRAEQUIPCEN 77-C-0096-1, April 1978.

13. G.R. Doddington, "Speaker Verification," Rome Air Development Center Technical Report, RADC-TR-74-179, April 1974, 785135/5GI.

14. J.S. Kenyon and T.A. Knott, *A Pronouncing Dictionary of American English*, G. & C. Merriam Company (Springfield, Massachusetts, 1953).

15. F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proceedings of the IEEE*, 64:532–556, April 1976.

16. N.R. Dixon and H.F. Silverman, "The 1976 Modular Acoustic Processor (MAP)," *IEEE Transactions on Acoustics, Speech and Signal Processing*, ASSP-25:367–379, October 1977.

17. F. Jelinek, L.R. Bahl, and R.L. Mercer, "Design of a Linguistic Statistical Decoder for the Recognition of Continuous Speech," *IEEE Transactions on Information Theory*, IT-21:250–256, May 1975.

18. L.R. Bahl and F. Jelinek, "Decoding for Channels With Insertions, Deletions, and Substitutions With Applications to Speech Recognition," *IEEE Transactions on Information Theory*, IT-21:404–411, July 1975.

19. L.R. Rabiner, "On Creating Reference Templates for Speaker Independent Recognition of Isolated Words," *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP-26:34–42, February 1978.

20. S.E. Levinson et al., "Interactive Clustering Techniques for Selecting Speaker-Independent Reference Templates for Isolated Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP-27:134–141, April 1979.

21. L.R. Rabiner et al., "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* 574–577, Washington, D.C., 2–4 April 1979.

22. L.R. Rabiner and J.G. Wilpon, "Considerations in Applying Clustering Techniques to Speaker-Independent Word Recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* 578–581, Washington, D.C., 2–4 April 1979.

23. K. Tanaka, "A Standard Category Pattern-Making Method With Application to Phoneme Recognition," *Proceedings of the Fourth International Joint Conference on Pattern Recognition,* 1030–1032, Kyoto, Japan, 7–10 November 1978.

24. K. Tanaka, "A Talker Clustering Method for Standard Pattern Making," *Progress Report on Speech Research '77,* Electrotechnical Laboratory, Japan, August 1978.

25. B. Gold, "Word-Recognition Computer Program," Massachusetts Institute of Technology, Cambridge, RLE Technical Report 452, June 1966.

26. L.R. Rabiner et al., "Considerations in Dynamic Time Warping Algorithms for Discrete Word Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing,* ASSP-26:575–582, December 1978.

27. K. Fukunaga and L.D. Hostetler, "The Estimation of the Gradient of a Density Function, With Applications in Pattern Recognition," *IEEE Transactions on Information Theory,* IT-21:32–40, January 1975.

28. R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis,* John Wiley and Sons (New York, 1973).

29. J.A. Hartigan, *Clustering Algorithms,* John Wiley and Sons (New York, 1975).

30. B. Everitt, *Cluster Analysis,* Heinemann Educational Books, Ltd. (London, 1974).

31. T. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis" (unpublished), 1971.

32. M.R. Anderberg, *Cluster Analysis for Applications,* Academic Press (New York, 1973).

33. F.M. Reza, *An Introduction to Information Theory,* McGraw-Hill Book Company (New York, 1961).

APPENDIX A
SPEECH PROCESSING

The speech processing used in this study is based on the relative spectrum of speech as a function of time, which is the output of a 16-channel digital filter bank that has been preprocessed as described in this appendix.

1. FILTER BANK DEFINITION

The spectrum is obtained by processing the speech signal through a digital filter bank preceded by a first-order differencing network (for preemphasis). The filter bank consists of 16 bandpass filters, each followed by a full-wave rectifier and a four-pole lowpass Bessel filter with a 3-dB cutoff at 30 Hz. Each of the 16 filters is sampled 100 times per second. A block diagram of the spectral analysis hardware is shown in Figure A-1. Actual filter responses appear in Figure A-2 for the bandpass filters alone and in Figure A-3 for the bandpass filters with preemphasis.

For processing, the top three filters are summed and filter 14 is replaced by this sum. Filters 15 and 16 are set to zero. The resulting 14 filter outputs at each time sample are represented by:

$$A_j = \begin{Bmatrix} a_{1j} \\ a_{2j} \\ \cdot \\ \cdot \\ \cdot \\ a_{14j} \end{Bmatrix} = \begin{Bmatrix} a_1(t_j) \\ a_2(t_j) \\ \cdot \\ \cdot \\ \cdot \\ a_{14}(t_j) \end{Bmatrix} \tag{A-1}$$

2. REGRESSION

It has been found that, by eliminating the gross aspects of the spectrum, such as the slope and curvature, more clearly defined formant frequencies are obtained. Therefore, the spectral amplitude vector is regressed by the first three elements of an orthonormal basis set:

$$(\vec{A}_j)_R = \vec{A}_j - \sum_{k=0}^{2} c_{jk} \vec{F}_k \tag{A-2}$$

where

$$\vec{F}_k = \begin{Bmatrix} f_{1k} \\ \cdot \\ \cdot \\ \cdot \\ f_{14k} \end{Bmatrix} \quad k = \{0, 1, 2\}$$

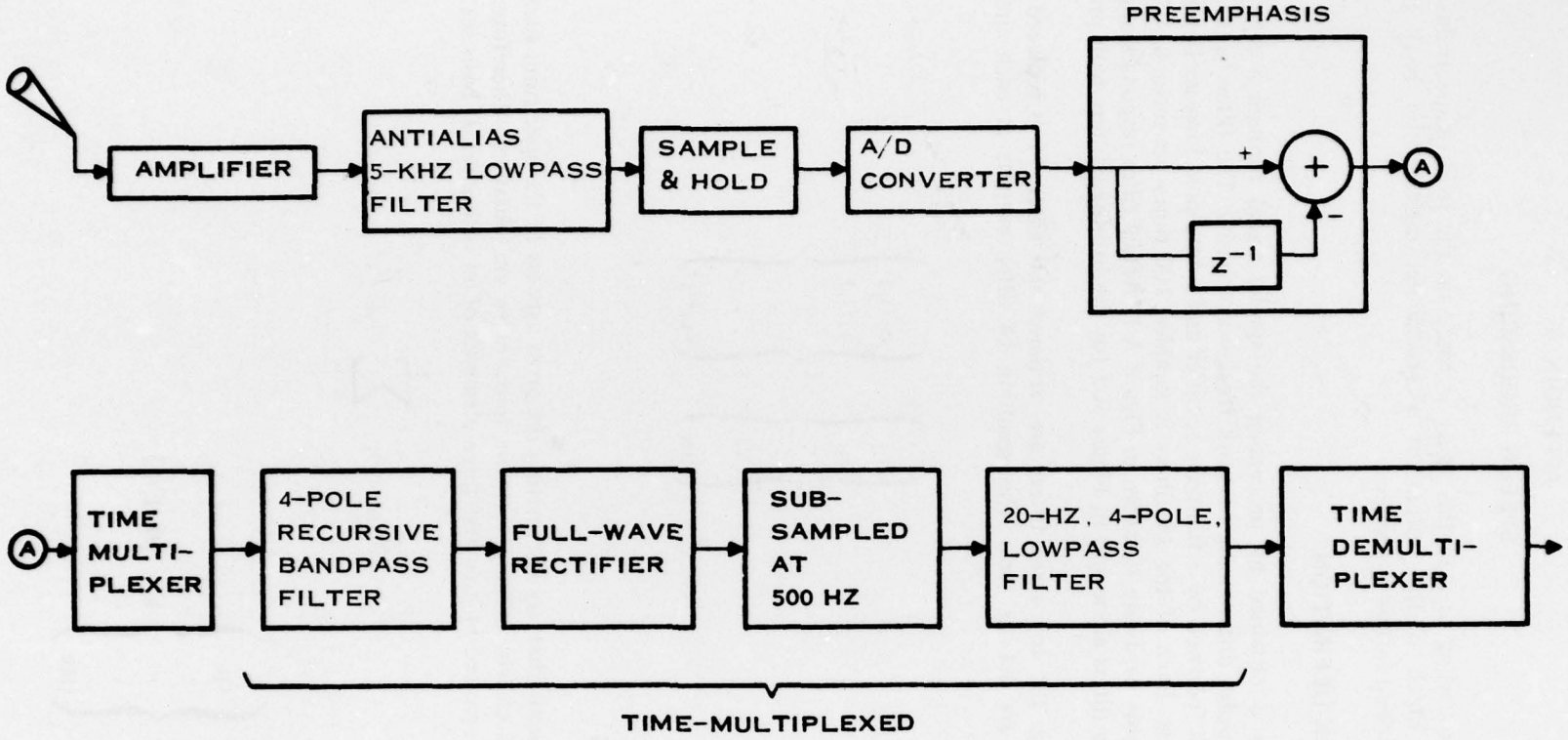


Figure A-1. Spectral Preprocessing Functional Block Diagram

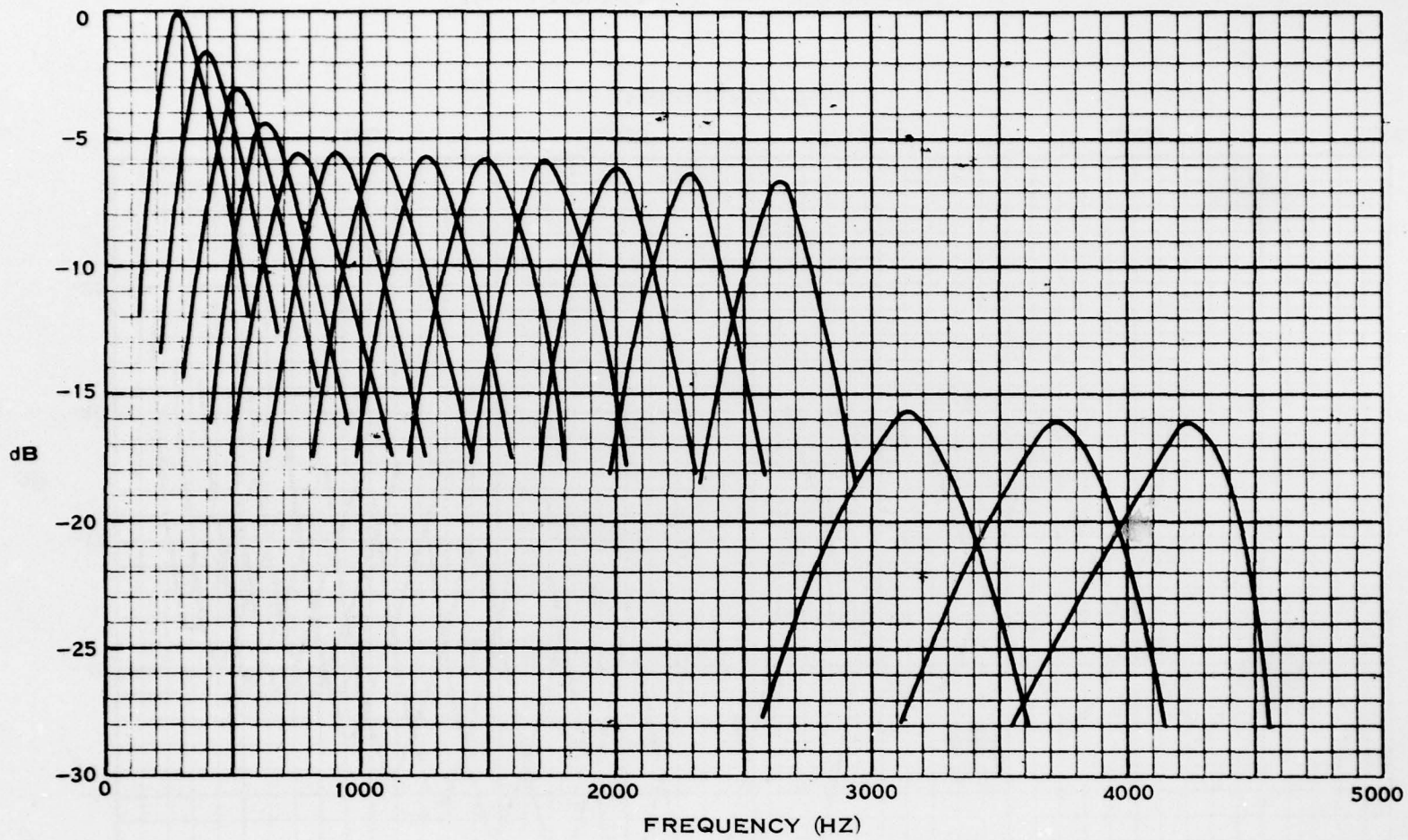


Figure A-2. Digital Filter Responses

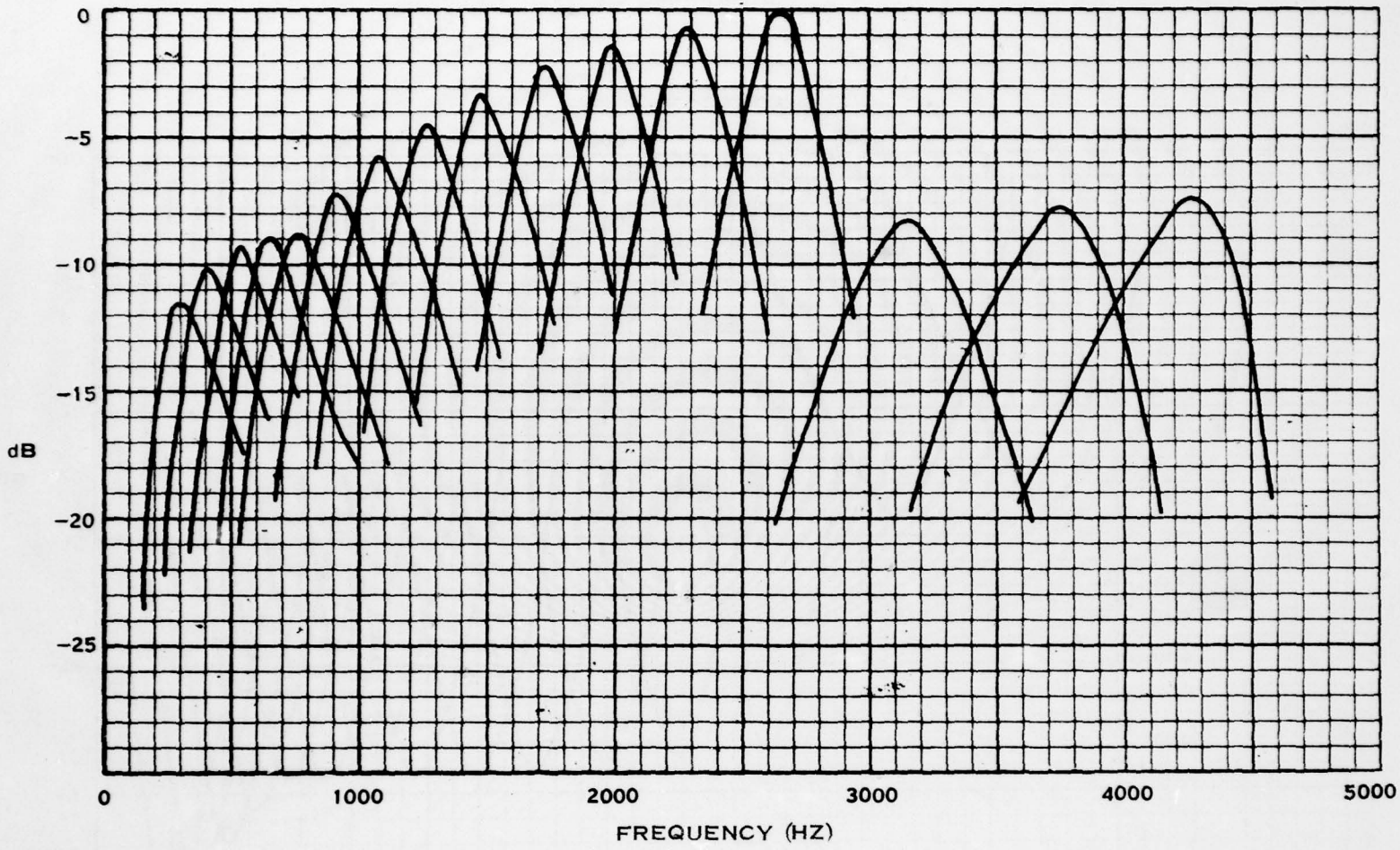


Figure A-3. Digital Filter Responses With Preemphasis

$$f_{i0} = \frac{1}{\sqrt{14}}$$

$$f_{i1} = -\frac{1}{\sqrt{7}} \sin\left[\frac{(i - \tfrac{1}{2})}{14}\pi\right] \qquad \{i = 1, 2, \ldots, 14\}$$

$$f_{i2} = -\frac{1}{\sqrt{7}} \cos\left[\frac{(i - \tfrac{1}{2})}{14}\pi\right]$$

and

$$c_{jk} = \sum_{m=1}^{14} a_{mj} f_{mk}$$

Thus, the regression tends to flatten the spectrum, removing any half-cycle sine or cosine wave trends of the spectrum at time t_j. An example of a spectral waveform having a large positive c_1 is a nasal, which has one peak near the low end and one near the high end of the spectrum (around 250 Hz and 2200 Hz). An example of a spectral waveform with a large positive c_2 is a sibilant, having most of its energy above 3000 Hz. Most vowels, however, have the opposite spectral tilt because of the glottal source spectral decay with increasing frequency, yielding a large negative value of c_2.

3. NORMALIZATION

The regressed amplitude vector is next normalized by a modified postregression standard deviation, σ_j^* for time t_j:

$$\sigma_j^* = \sigma_{post_j} + \sigma_{min} \tag{A-3}$$

where

$$\sigma^2_{post_j} = \frac{1}{11}\left(\sum_{m=1}^{14} a_{mj}{}^2 - \sum_{k=0}^{2} c_{jk}^2\right)$$

and $\sigma_{min} = 62$ for this study. However, it has been noticed that regression sometimes eliminates too much of the variance of the filter output vector A_j. To limit the regression, a limit is placed on σ_{post} as follows:

$$\sigma_{post_j} = \max\left(\sigma^2_{post_j}, R^2_{min}\ \sigma^2_{pre_j}\right) \tag{A-4}$$

where

$$\sigma^2_{pre_j} = \frac{1}{13}\left(\sum_{m=1}^{14} a_{mj}^2 - c_{j0}^2\right)$$

and $R_{min} = 0.6$. Note that, when $\sigma_{post_j} = R_{min}\ \sigma_{pre_j}$, the regression coefficients c_1 and c_2 are reduced in order to decrease the amount of regression. The resulting normalized amplitude vector is:

$$(\vec{A}_j)_N = \frac{1}{\sigma_j^*} (\vec{A}_j)_R \tag{A-5}$$

The regression coefficients c_1 and c_2 are also normalized by σ_j^*.

4. QUANTIZATION

The regressed and normalized amplitude vector is then quantized to one of eight levels according to a set of quantization thresholds ϕ_{iq} :

$$(a_{ij})_Q = q \text{ IFF } \begin{cases} (a_{ij})_N \geq \phi_{iq} \\ \\ (a_{ij})_N < \phi_{i,q+1} \text{ for } q = 0, 1, \ldots, 7 \end{cases} \tag{A-6}$$

where $\phi_{iq} < \phi_{i,q+1}$; $\phi_{i0} = -\infty$; and $\phi_{i8} = \infty$.

Rather than have these quantization levels (ϕ_{iq}) being chosen to yield a uniform probability, however, it was more desirable to have the quantization thresholds cluster at higher energy levels. In this way, the sensitivity to noise can be reduced and quantization resolution is increased in the region of interest (which is the spectrum amplitude at the formant frequencies). The actual procedure used to determine the quantization thresholds is described in more detail in the total voice verification study final report;[1] however, the quantization thresholds used for each of the 14 filter outputs are shown in Figure A-4 and those used for the two regression coefficients c_1 and c_2 are shown below:

	ϕ_0	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8
c_1	$-\infty$	−3.0	−1.5	0	1.5	3.0	4.5	6.0	∞
c_2	$-\infty$	−7.0	−5.67	−4.33	−3	−1.67	−0.33	1.0	∞

5. ENERGY

For each time sample, a measure of the energy was also computed. As an aid to distinguishing vowels from nasals (which usually have most of their energy in a_{1_j}) and vowels from sibilants (which usually have most of their energy in a_{14_j}), these two filters were not used in computing the energy measure in the following expression:

$$E = \sqrt{\sum_{i=2}^{13} (a_i)^2 - \frac{1}{11}\left(\sum_{i=2}^{13} a_i\right)^2} \tag{A-7}$$

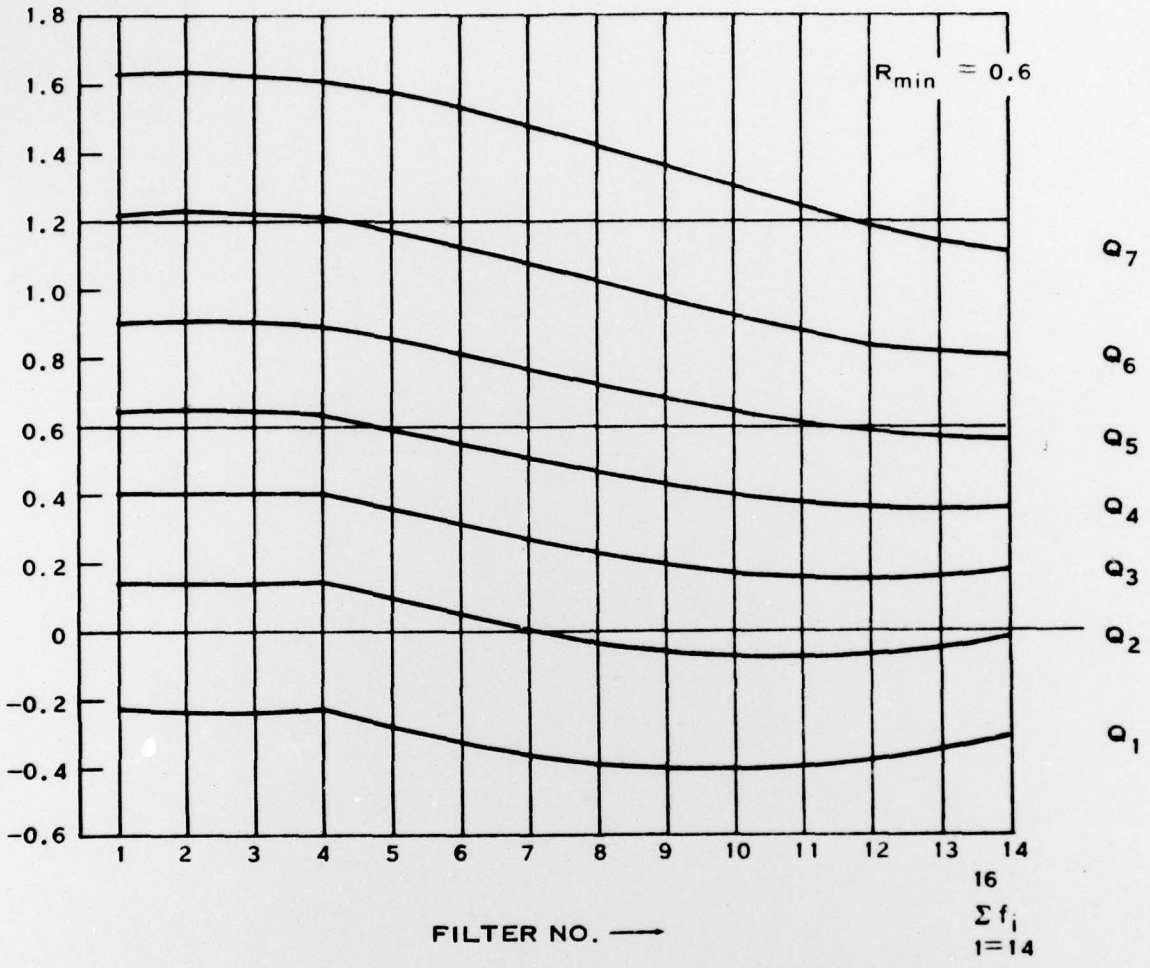


Figure A-4. Quantization Thresholds for Filters

APPENDIX B
SCATTER MATRICES

Inherent in many criteria used in clustering is the concept of the scatter of the data, numerically represented by scatter matrices. The within-class scatter matrix measures the distance of the "n" sample vectors from their mean vectors and is the sum of the scatter matrices for all "c" classes. The within-class scatter is given by

$$S_W = \sum_{i=1}^{c} S_i = \sum_{i=1}^{c} \sum_{x \in X_i} (\vec{x} - \vec{m}_i)(\vec{x} - \vec{m}_i)^T \qquad \text{(B-1)}$$

The between-class scatter matrix measures the distance of the class mean vectors from the overall sample mean and is given by

$$S_B = \sum_{i=1}^{c} n_i (\vec{m}_i - \vec{m})(\vec{m}_i - \vec{m})^T \qquad \text{(B-2)}$$

The total scatter matrix measures the distance of the samples from the overall mean as given by

$$S_T = \sum_{x \in X} (\vec{x} - \vec{m})(\vec{x} - \vec{m})^T \qquad \text{(B-3)}$$

Note that $S_T = S_W + S_B$, and, therefore, $|S_T| = |S_W| + |S_B|$ and $\text{tr } S_T = \text{tr } S_W + \text{tr } S_B$.

Anderberg[32] summarizes the four principal criteria that have emerged using scatter matrices:

(1) Minimize $\text{tr } S_W$. This is identical to the sum-of-squared error criterion of Ward.

(2) Minimize the ratio $|S_W|/|S_T|$. This criterion is known as Wilks' lambda statistic. Equivalent criteria are minimizing $|S_W|$ or maximizing $|S_T|/|S_W|$ or $|I + S_W^{-1} S_B|$.

(3) Maximize largest eigenvalue of $S_W^{-1} S_B$ (attributed to S.N. Roy).

(4) Maximize $\text{tr } S_W^{-1} S_B$ (Hotelling's trace criterion).

The last three of these criteria involve the eigenvalues of $S_W^{-1} S_B$ that are invariant under nonsingular linear transformations, measuring the ratio of the between-class to within-class scatter in the direction of the eigenvectors. Duda and Hart,[28] however, note that invariant criterion functions are more likely to possess multiple local extrema, and are correspondingly more difficult to extremize.

As noted, the first criterion is simply the sum-of-squared error criterion,

$$J_e = \sum_{i=1}^{c} J_i$$

PRECEDING PAGE NOT FILMED
BLANK

Specifically,

$$J_c = \text{tr } S_W = \sum_{i=1}^{c} \text{tr } S_i = \sum_{i=1}^{c} \sum_{\vec{x} \in X_i} \|\vec{x} - \vec{m}_i\|^2 \tag{B-4}$$

With some manipulation, this can be rewritten as

$$J_c = \tfrac{1}{2} \sum_{i=1}^{c} \frac{1}{n_i} \sum_{\vec{x} \in X_i} \sum_{\vec{x}' \in X_i} \|\vec{x} - \vec{x}'\|^2 \tag{B-5}$$

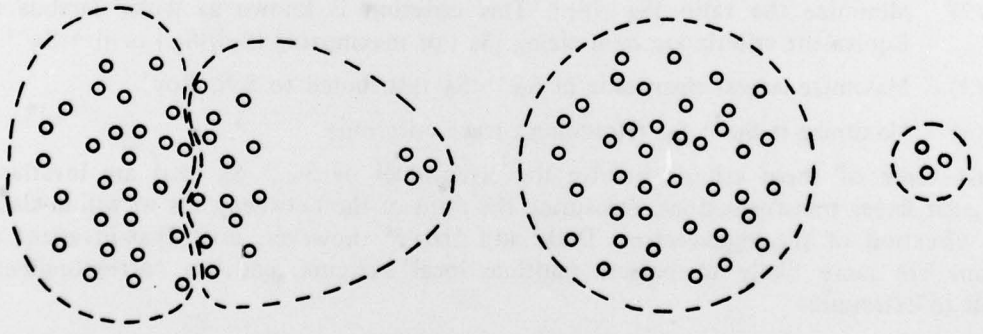
Minimization of this criterion is equivalent to maximizing

$$\text{tr } S_B = \sum_{i=1}^{c} n_i \|\vec{m}_i - \vec{m}\|^2 \tag{B-6}$$

since tr S_T is a constant. The term tr S_B can also be manipulated (see Appendix C) to give

$$\text{tr } S_B = \frac{1}{2n} \sum_{i=1}^{c} \sum_{j=1}^{c} n_i n_j \|\vec{m}_i - \vec{m}_j\|^2 \tag{B-7}$$

Although the criteria given above differ, the underlying model using scatter matrices is that of "c" fairly well separated clouds containing roughly equal numbers of points. Duda and Hart,[28] however, demonstrate (Figure B-1) how this model can lead to poor results.



(A) (B)

SUM-OF-SQUARED ERROR IS SMALLER FOR (A) THAN FOR (B)
(FROM DUDA AND HART[28])

Figure B-1. Problem of Splitting Large Clusters

APPENDIX C

tr S_B ALTERNATE FORM DERIVATION

By definition,

$$\text{tr } S_B = \sum_{i=1}^{c} n_i \, \|\vec{m}_i - \vec{m}\|^2 = \sum_{i=1}^{c} n_i \, (\vec{m}_i - \vec{m})^T (\vec{m}_i - \vec{m}) \tag{C-1}$$

Expanding,

$$\text{tr } S_B = \sum_{i=1}^{c} n_i \vec{m}_i^T \vec{m}_i - 2 \sum_{i=1}^{c} n_i \vec{m}_i^T \vec{m} + \sum_{i=1}^{c} n_i \vec{m}^T \vec{m} \tag{C-2}$$

But the sum in the middle term can be rewritten as

$$\sum_{i=1}^{c} n_i \vec{m}_i^T = n \vec{m}^T = \sum_{i=1}^{c} n_i \, \vec{m}^T \tag{C-3}$$

yielding

$$\text{tr } S_B = \sum_{i=1}^{c} n_i \vec{m}_i^T \vec{m}_i - \sum_{i=1}^{c} n_i \vec{m}^T \vec{m} \tag{C-4}$$

Multiplying by (2n/2n) and splitting the first term in half with appropriate change in indices gives

$$\text{tr } S_B = (1/2n) \left[n \sum_{i=1}^{c} n_i \vec{m}_i^T \vec{m}_i - 2 \sum_{i=1}^{c} n_i m_i^T \sum_{j=1}^{c} n_j \vec{m}_j + n \sum_{j=1}^{c} n_j \vec{m}_j^T \vec{m}_j \right] \tag{C-5}$$

Expressing n as the sum of the $n_i s$ (or $n_j s$) and factoring out the summations and $n_i n_j$ yields

$$\text{tr } S_B = (1/2n) \sum_{i=1}^{c} \sum_{j=1}^{c} n_i n_j \left[\vec{m}_i^T \vec{m}_i - 2 \, \vec{m}_i^T \vec{m}_j + \vec{m}_j^T \vec{m}_j \right] \tag{C-6}$$

$$\text{tr } S_B = (1/2n) \sum_{i=1}^{c} \sum_{j=1}^{c} n_i n_j \|\vec{m}_i - \vec{m}_j\|^2 \qquad \text{(C-7)}$$

APPENDIX D

POST ITERATIVE OPTIMIZATION STATISTICS FOR RECOGNITION PATTERNS

STATISTICS FOR DIGIT: 0; REF PT: 0; NO OF DATA PTS: 166
MINAVE AND MINMAX AGGLOM CLUSTERING; 16 MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	69662.7	69662.7	0.148	0.148	0.000	0.000
2	75	51	59320.9	59345.7	0.058	0.037	0.174	0.174
3	97	34	55867.7	57150.6	0.034	0.053	0.247	0.219
4	124	42	53969.2	54138.4	0.029	0.040	0.291	0.287
5	134	40	52407.1	51966.8	0.022	0.037	0.329	0.341
6	113	50	51245.6	50045.1	0.047	0.019	0.359	0.392
7	161	60	48820.9	49112.9	0.012	0.017	0.427	0.418
8	136	52	48248.7	48258.7	0.016	0.020	0.444	0.444
9	116	47	47489.6	47307.3	0.027	0.027	0.467	0.473
10	122	85	46223.7	46051.3	**********		0.507	0.513

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	24.495	24.436	0.000	0.000
2	0.172	0.172	0.343	0.343	9.547	6.066	28.591	28.511
3	0.182	0.161	0.372	0.319	5.539	8.591	20.124	17.843
4	0.189	0.187	0.401	0.359	4.689	6.498	15.702	15.485
5	0.200	0.206	0.436	0.433	3.568	5.954	13.253	13.706
6	0.208	0.227	0.471	0.490	7.570	2.980	11.500	12.544
7	0.239	0.234	0.501	0.496	1.864	2.765	11.313	11.088
8	0.241	0.241	0.524	0.502	2.486	3.115	10.018	10.011
9	0.248	0.251	0.583	0.641	4.185	4.168	9.163	9.274
10	0.265	0.268	0.609	0.604	**********		8.789	8.887

STATISTICS FOR DIGIT: 1; REF PT: 0; NO OF DATA PTS: 168
MINAVE AND MINMAX AGGLOM CLUSTERING; 16 MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	70679.1	70679.1	0.137	0.138	0.000	0.000
2	19	33	60969.0	60914.4	0.064	0.077	0.159	0.160
3	73	58	57069.9	56197.6	0.078	0.063	0.238	0.258
4	86	57	52645.4	52655.6	0.031	0.037	0.343	0.342
5	132	75	50998.3	50683.9	0.033	0.026	0.386	0.395
6	135	74	49329.7	49352.1	0.033	0.039	0.433	0.432
7	155	81	47704.8	47402.8	0.031	0.023	0.482	0.491
8	112	91	46239.0	46291.0	0.025	0.028	0.529	0.527
9	132	93	45093.1	44985.8	0.023	0.017	0.567	0.571
10	166	83	44062.9	44201.1	**********		0.604	0.599

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	22.943	23.072	0.000	0.000
2	0.157	0.158	0.317	0.318	10.616	12.854	26.438	26.610
3	0.176	0.190	0.359	0.379	12.792	10.400	19.673	21.259
4	0.223	0.223	0.464	0.506	5.131	6.141	18.726	18.712
5	0.234	0.239	0.495	0.520	5.333	4.283	15.726	16.076
6	0.250	0.250	0.533	0.535	5.336	6.399	14.022	14.001
7	0.269	0.275	0.584	0.560	4.947	3.776	12.923	13.176
8	0.288	0.287	0.590	0.567	3.965	4.511	12.081	12.042
9	0.302	0.304	0.621	0.625	3.633	2.773	11.277	11.351
10	0.316	0.313	0.630	0.674	**********		10.604	10.516

STATISTICS FOR DIGIT: 2; REF PT: 0; NO OF DATA PTS: 166
MINAVE AND MINMAX AGGLOM CLUSTERING; 16 MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	51525.5	51525.5	0.125	0.124	0.000	0.000
2	72	48	45108.0	45140.0	0.061	0.060	0.142	0.141
3	105	48	42371.1	42424.6	0.047	0.030	0.216	0.215
4	140	50	40377.6	41131.9	0.027	0.049	0.276	0.253
5	132	82	39269.9	39098.7	0.034	0.035	0.312	0.318
6	92	124	37929.8	37788.9	0.025	0.027	0.358	0.364
7	116	121	36991.1	36777.1	0.031	0.031	0.393	0.401
8	134	133	35861.8	35640.8	0.015	0.015	0.437	0.446
9	102	113	35323.3	35122.3	0.032	0.014	0.459	0.467
10	138	83	34182.5	34613.7	**********		0.507	0.489

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	20.675	20.572	0.000	0.000
2	0.140	0.139	0.285	0.284	10.011	9.926	23.474	23.341
3	0.158	0.157	0.328	0.329	7.716	4.997	17.716	17.591
4	0.179	0.163	0.363	0.437	4.472	8.057	15.001	15.729
5	0.188	0.192	0.385	0.486	5.528	5.427	12.640	12.872
6	0.206	0.209	0.496	0.494	3.985	4.311	11.542	11.705
7	0.218	0.223	0.545	0.499	4.885	4.943	10.478	10.694
8	0.236	0.241	0.549	0.532	2.388	2.313	9.921	10.123
9	0.243	0.247	0.548	0.539	5.102	2.288	9.059	9.224
10	0.263	0.254	0.608	0.626	**********		8.851	8.523

STATISTICS FOR DIGIT: 3; REF PT: 0; NO OF DATA PTS: 169
MINAVE AND MINMAX AGGLOM CLUSTERING; 16 MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	54075.5	54075.5	0.118	0.118	0.000	0.000
2	70	51	47673.7	47698.1	0.056	0.056	0.134	0.134
3	98	70	44988.2	45016.6	0.042	0.045	0.202	0.201
4	113	49	43089.4	43007.2	0.050	0.040	0.255	0.257
5	118	67	40949.0	41295.5	0.035	0.038	0.321	0.309
6	90	68	39523.6	39708.2	0.017	0.042	0.368	0.362
7	76	66	38841.5	38043.4	0.033	0.031	0.392	0.421
8	106	58	37565.5	36879.2	0.020	0.024	0.439	0.466
9	112	82	36810.8	36009.3	0.031	0.021	0.469	0.502
10	127	67	35675.6	35267.3	**********		0.516	0.533

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	19.889	19.813	0.000	0.000
2	0.133	0.132	0.270	0.270	9.407	9.388	22.425	22.328
3	0.149	0.148	0.302	0.295	7.006	7.410	16.765	16.702
4	0.166	0.168	0.365	0.348	8.196	6.567	14.023	14.155
5	0.194	0.188	0.392	0.382	5.709	6.304	13.143	12.688
6	0.213	0.209	0.458	0.435	2.813	6.834	12.003	11.795
7	0.219	0.236	0.493	0.490	5.322	4.957	10.590	11.378
8	0.239	0.254	0.566	0.547	3.234	5.798	10.108	10.725
9	0.250	0.267	0.594	0.550	4.934	5.297	9.380	10.034
10	0.270	0.279	0.639	0.579	**********		9.112	9.422

STATISTICS FOR DIGIT: 4; REF PT: 0; NO OF DATA PTS: 167
MINAVE AND MINMAX AGGLOM CLUSTERING; 20MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	43604.4	43604.4	0.127	0.127	0.000	0.000
2	99	32	38061.0	38075.8	0.054	0.053	0.146	0.145
3	84	64	35993.5	36052.4	0.040	0.036	0.211	0.209
4	133	59	34536.4	34738.5	0.029	0.040	0.263	0.255
5	124	123	33525.7	33340.8	0.034	0.039	0.301	0.308
6	113	107	32389.8	32054.9	0.041	0.031	0.346	0.360
7	128	116	31074.0	31056.6	0.015	0.025	0.403	0.404
8	141	83	30617.9	30289.4	0.015	0.035	0.424	0.440
9	149	129	30165.9	29229.9	0.034	0.011	0.445	0.492
10	200	115	29153.2	28899.5	**********		0.496	0.509

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	20.468	20.413	0.000	0.000
2	0.140	0.139	0.293	0.295	8.691	8.503	23.303	23.232
3	0.151	0.150	0.375	0.365	6.437	5.795	16.810	16.653
4	0.166	0.161	0.353	0.385	4.623	6.357	13.828	13.442
5	0.177	0.181	0.461	0.452	5.320	6.055	11.800	12.083
6	0.194	0.202	0.473	0.532	6.337	4.858	10.803	11.241
7	0.218	0.219	0.541	0.527	2.275	3.829	10.417	10.437
8	0.224	0.232	0.583	0.539	2.273	5.387	9.331	9.671
9	0.230	0.253	1.044	0.590	5.137	1.730	8.520	9.405
10	0.251	0.257	0.645	0.611	**********		8.372	8.594

STATISTICS FOR DIGIT: 5; REF PT: 0; NO OF DATA PTS: 169
MINAVE AND MINMAX AGGLOM CLUSTERING; 20MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	87861.3	87861.3	0.164	0.164	0.000	0.000
2	15	19	73448.7	73443.4	0.041	0.053	0.196	0.196
3	52	56	70442.7	69540.9	0.050	0.047	0.247	0.263
4	85	45	66955.3	66296.3	0.045	0.032	0.312	0.325
5	108	46	63947.1	64185.0	0.028	0.026	0.374	0.369
6	127	43	62144.9	62522.0	0.025	0.029	0.414	0.405
7	126	59	60578.4	60714.5	0.034	0.025	0.450	0.447
8	150	68	58523.0	59195.0	0.024	0.030	0.501	0.484
9	167	75	57107.8	57430.3	0.021	0.023	0.539	0.530
10	178	84	55926.8	56100.6	**********		0.571	0.566

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	27.394	27.404	0.000	0.000
2	0.193	0.193	0.390	0.390	6.794	8.820	32.574	32.588
3	0.181	0.193	0.335	0.368	8.169	7.699	20.400	21.734
4	0.202	0.210	0.383	0.403	7.368	5.223	17.069	17.782
5	0.225	0.222	0.441	0.414	4.594	4.223	15.239	15.032
6	0.238	0.233	0.467	0.438	4.084	4.683	13.408	13.131
7	0.250	0.248	0.509	0.478	5.463	4.030	12.085	11.998
8	0.271	0.262	0.540	0.542	3.869	4.770	11.459	11.069
9	0.285	0.280	0.561	0.577	3.288	3.681	10.703	10.531
10	0.297	0.294	0.611	0.597	**********		10.024	9.939

STATISTICS FOR DIGIT: 6; REF PT: 0; NO OF DATA PTS: 167
MINAVE AND MINMAX AGGLOM CLUSTERING; 20MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	58412.0	58412.0	0.176	0.176	0.000	0.000
2	84	47	48111.0	48136.5	0.051	0.051	0.214	0.213
3	84	27	45669.2	45686.7	0.027	0.055	0.279	0.279
4	68	42	44419.0	43177.7	0.059	0.022	0.315	0.353
5	88	42	41809.1	42246.4	0.025	0.030	0.397	0.383
6	100	47	40753.9	40986.8	0.035	0.019	0.435	0.425
7	133	49	39326.5	40228.1	0.017	0.036	0.485	0.452
8	130	90	38659.4	38789.0	0.028	0.003	0.511	0.506
9	160	32	37594.0	38654.9	0.016	0.021	0.554	0.511
10	145	60	36996.0	37826.5	**********		0.579	0.544

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	28.921	28.850	0.000	0.000
2	0.209	0.208	0.422	0.422	8.273	8.296	54.900	34.795
3	0.203	0.203	0.430	0.426	4.435	8.897	22.601	22.561
4	0.202	0.227	0.371	0.518	9.460	5.473	16.906	18.955
5	0.238	0.229	0.475	0.530	4.038	4.771	15.864	15.306
6	0.248	0.243	0.519	0.486	5.569	2.943	13.779	15.520
7	0.268	0.249	0.565	0.546	2.680	5.652	12.780	11.905
8	0.274	0.272	0.572	0.601	4.327	0.543	11.460	11.546
9	0.291	0.269	0.590	0.601	2.481	3.343	10.798	9.967
10	0.298	0.281	0.608	0.633	**********		9.969	9.372

STATISTICS FOR DIGIT: 7; REF PT: 0; NO OF DATA PTS: 168
MINAVE AND MINMAX AGGLOM CLUSTERING; 20MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	80124.8	80124.8	0.194	0.193	0.000	0.000
2	81	2	64595.8	64656.4	0.061	0.064	0.240	0.239
3	32	17	60652.0	60524.4	0.037	0.030	0.321	0.324
4	71	36	58433.6	58724.0	0.020	0.058	0.371	0.364
5	60	67	57262.1	55331.3	0.043	0.029	0.399	0.448
6	85	72	54819.4	53704.3	0.018	0.029	0.462	0.492
7	93	62	53809.7	52160.0	0.042	0.028	0.489	0.536
8	106	79	51550.4	50680.3	0.033	0.012	0.554	0.561
9	104	45	49846.8	50085.5	0.021	0.015	0.607	0.600
10	98	43	48808.2	49334.1	**********		0.642	0.624

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	32.172	32.047	0.000	0.000
2	0.236	0.235	0.475	0.475	10.074	10.545	59.666	39.474
3	0.235	0.237	0.456	0.467	5.998	4.878	26.327	26.555
4	0.240	0.236	0.451	0.485	3.268	9.417	20.169	19.801
5	0.241	0.270	0.475	0.532	6.911	4.763	16.170	18.148
6	0.265	0.283	0.545	0.568	2.965	4.630	14.864	15.641
7	0.272	0.298	0.565	0.659	6.718	4.539	13.041	14.297
8	0.300	0.314	0.655	0.671	5.254	1.860	12.591	13.197
9	0.321	0.317	0.727	0.672	3.292	2.570	11.997	11.845
10	0.353	0.324	0.717	0.814	**********		11.193	10.898

STATISTICS FOR DIGIT: 8; REF PT: 0; NO OF DATA PTS: 168
MINAVE AND MINMAX AGGLOM CLUSTERING; 20MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	61486.2	61486.2	0.166	0.166	0.000	0.000
2	90	13	51255.8	51286.6	0.071	0.071	0.200	0.199
3	135	26	47621.8	47656.7	0.080	0.081	0.291	0.290
4	109	50	43789.9	43799.9	0.031	0.032	0.404	0.404
5	100	63	42426.6	42403.3	0.024	0.028	0.449	0.450
6	102	55	41395.2	41195.1	0.029	0.027	0.485	0.493
7	133	48	40203.8	40101.5	0.018	0.029	0.529	0.533
8	125	52	39474.0	38918.9	0.041	0.022	0.558	0.580
9	140	44	37863.9	38054.1	0.029	0.043	0.624	0.616
10	123	58	36747.5	36408.3	**********		0.673	0.689

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	27.786	27.703	0.000	0.000
2	0.197	0.197	0.394	0.394	11.769	11.749	33.133	33.013
3	0.214	0.214	0.429	0.429	13.277	13.353	24.019	23.941
4	0.263	0.263	0.528	0.528	5.106	5.229	22.092	22.074
5	0.272	0.275	0.558	0.571	5.963	4.644	18.306	18.339
6	0.281	0.285	0.571	0.579	4.662	4.301	15.725	15.959
7	0.296	0.298	0.657	0.628	2.922	4.748	14.205	14.309
8	0.303	0.316	0.628	0.669	6.527	3.555	12.746	13.254
9	0.332	0.328	0.675	0.743	4.688	6.676	12.400	12.238
10	0.352	0.360	0.754	0.805	**********		11.819	12.092

STATISTICS FOR DIGIT: 9; REF PT: 0; NO OF DATA PTS: 169
MINAVE AND MINMAX AGGLOM CLUSTERING; 20MAR79
POST ITERATIVE OPTIMIZATION FOR MIN JE

C	NO OF ITERS		JE (=TR(W))		JE(C)-JE(C+1) /JE(C)		TR(B)/TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0	0	75929.8	75929.8	0.134	0.134	0.000	0.000
2	74	15	65731.1	65777.5	0.060	0.062	0.155	0.154
3	81	27	61771.8	61677.0	0.053	0.051	0.229	0.231
4	128	46	58517.1	58556.2	0.057	0.057	0.298	0.297
5	150	110	55207.2	55211.1	0.049	0.030	0.375	0.375
6	140	62	52482.7	53535.6	0.026	0.046	0.447	0.418
7	117	110	51106.2	51086.3	0.026	0.010	0.486	0.486
8	121	58	49760.9	50559.5	0.047	0.044	0.526	0.502
9	107	70	47428.2	48322.0	0.022	0.025	0.601	0.571
10	141	76	46395.2	47105.9	**********		0.637	0.612

C	C(N-C)TR(B) /2N(C-1)TR(W)		BTL'S SIGMA		(N-C)*DELJE /JE(C)		(N-C)*TR(B) /(C-1)*TR(W)	
	AVE	MAX	AVE	MAX	AVE	MAX	AVE	MAX
1	0.000	0.000	0.000	0.000	22.565	22.463	0.000	0.000
2	0.153	0.153	0.307	0.307	10.059	10.411	25.911	25.775
3	0.169	0.170	0.345	0.342	8.746	8.399	19.023	19.180
4	0.194	0.193	0.392	0.392	9.333	9.426	16.366	16.318
5	0.228	0.228	0.477	0.458	8.093	4.977	15.390	15.386
6	0.259	0.242	0.536	0.497	4.275	7.457	14.564	13.637
7	0.272	0.272	0.549	0.582	4.264	1.671	13.115	13.130
8	0.286	0.273	0.611	0.597	7.547	7.125	12.096	11.541
9	0.320	0.304	0.685	0.657	3.485	4.026	12.019	11.427
10	0.333	0.320	0.721	0.728	**********		11.246	10.810

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.