On the Application of Embedded Digit Training to Speaker Independent Connected Digit Recognition

LAWRENCE R. RABINER, FELLOW, IEEE, JAY G. WILPON, ANN M. QUINN, AND SANDRA G. TERRACE

Abstract—In recent years, several algorithms have been proposed for recognizing a string of connected words (typically digits) by optimally piecing together reference patterns corresponding to the words in the string. Although the algorithms differ greatly in details of implementation, storage requirements, etc., they all have essentially the same performance in that their ability to match the unknown string is related to how well words spoken in isolation can match their counterparts in connected speech. For low rates of articulation (i.e., about 100–130 words per minute) the performance of such connected word recognition systems is excellent. However, as the articulation rate approaches that of continuous discourse (180–300 words per minute) the performance of such connected word recognizers falls dramatically. To partially alleviate these problems a modified training procedure was devised in which multiple versions of each reference word were used. The multiple versions included an isolated form for each word, and 2 versions of the word extracted from the middle of 3 word sequences. One of these embedded reference patterns represented a noncontextual token of the word (i.e., spoken in a format where the words on either side had minimal effect on the acoustic properties at the boundaries), and the second represented a highly contextual token of the word. It was shown that a training algorithm could be devised to obtain these embedded reference tokens, and that when using the multiple reference patterns, the performance in a speaker trained system was greatly improved at faster talking rates. In this paper we show how the embedded training technique can be extended to the case of speaker, independent connected word recognizers. In particular, we show that improved recognition performance on connected digit strings is obtained by using standard clustering procedures on the embedded tokens to give a speaker-independent embedded reference set. We also show that the use of the K-nearest neighbor (KNN) rule leads to additional real improvements in performance for recognizing strings of connected digits. A discussion of the types of problems that remain is given.

I. INTRODUCTION

THE state of the art in speech recognition technology currently supports modest systems for isolated word recognition (both speaker trained and speaker independent) [1]-[4], as well as some fairly sophisticated systems for connected word recognition (generally only speaker trained systems have been available to date) [5]-[9]. As digital hardware becomes more powerful and the cost of computation goes steadily down, more powerful recognition systems will become available for handling tasks involved in speaker independent connected word recognition, as well as various conversational mode recognizers which are currently implemented as laboratory demonstration systems [10]. In the research laboratory, efforts are directed at both maintaining and improving performance of existing recognizers, and at devising new and powerful recognition algorithms that can be combined with existing algorithms to increase reliability, robustness, and performance.

One of the current areas of most interest and activity is that of designing algorithms for recognizing connected strings of words (generally digits) in either a speaker trained or a speaker independent manner [6]-[9]. A wide range of algorithms for performing connected word recognition by piecing together individual reference patterns to find the best match to the unknown test string have been proposed. These algorithms, although differing greatly in implementation, all yield the same recognition performance in that they are fundamentally limited by the ability of isolated word reference patterns to match the same words spoken in context. For slow rates of articulation (i.e., in the range 100–130 words per minute) the recognition performance is high since the words in context are generally quite similar to the isolated word patterns used as references. However, for faster rates of articulation (i.e., in the range 150–200 words per minute) the recognition performance degrades due to the contextual differences in words which are no longer well matched by isolated reference patterns. To alleviate this problem, a modified training procedure was proposed [11], in which embedded word patterns were extracted from 3-word training sequences, in which the desired word was the middle word, and used in combination with the isolated word patterns in the recognizer. It was shown that in a speaker trained environment, the modified training procedure was capable of giving a high performance digit recognizer for almost any rate of talking.

In this paper, we show how the modified training procedure can be combined with the standard speaker independent clustering package to give a speaker independent set of embedded training patterns for use in a speaker independent connected digit recognizer. An evaluation of the connected digit recognizer, using 19 talkers, each speaking 40 strings of digits of varying length and at varying rates, gave string error rates of 10.3 percent for deliberately spoken strings (4.3 percent if the length of the digit string was known), and 7.4 percent for normally spoken strings (5.9 percent if the length of the digit string was known). An analysis of the types of errors that occurred and some possible ways of handling them is given in the text.

The outline of this paper is as follows. In Section II we briefly review the connected word recognition algorithm, and explain the embedded word training procedure. In Section III

Manuscript received May 2, 1983; revised September 22, 1983.
The authors are with ATT, Bell Laboratories, Murray Hill, NJ 07974.

0096-3518/84/0400-0272$01.00

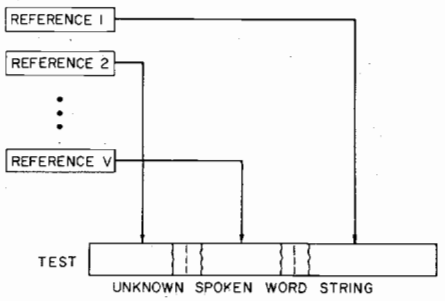


Fig. 1. Illustration of connected word recognition by concatenation of individual reference patterns.

we give results of an experimental evaluation of the recognizer for a digits vocabulary, and discuss the effects of several K-nearest neighbor decision rules. Finally, in Section IV we discuss the results and give some thoughts as to how the remaining problems could be handled in a practical system.

II. Review of the Connected Word Recognizer

The basic approach in a pattern-based approach to connected word recognition is summarized in Fig. 1. Assume we are given a test pattern, $\boldsymbol{T}$, which represents an unknown spoken word string, and we are given a set of V reference patterns, $\{R_1, R_2, \cdots, R_v\}$, each representing some word of the vocabulary. The connected word recognition problem consists of finding the "super" reference pattern, $\boldsymbol{R}^s$,

$$\boldsymbol{R}^s = R_{q(1)} \oplus R_{q(2)} \oplus \cdots \oplus R_{q(L)} \tag{1}$$

which is the concatenation of L reference patterns, $R_{q(1)}$, $R_{q(2)}, \cdots, R_{q(L)}$, which best matches the test string $\boldsymbol{T}$, in the sense that the overall distance between $\boldsymbol{T}$ and $\boldsymbol{R}^s$ is minimum over all possible choices of $L, q(1), q(2), \cdots, q(L)$, where the distance is an appropriately chosen distance measure.

There are several problems associated with solving the above connected word recognition problem. First, we don't know L, the number of words in the word string. Hence, our proposed solution must provide the best matches for all reasonable values of L, e.g., $L = 1, 2, \cdots, L_{\max}$. Second, we don't know nor can we reliably find word boundaries, even when we have postulated L, the number of words in the string. The implication of this observation is that our word recognition algorithm must work without direct knowledge of word boundaries; in fact the estimated word boundaries will be shown to be a by-product of the matching procedure. The third problem with a template matching approach is that the word matches are generally much poorer at the boundaries then at frames within the word. In general, this is a weakness of word matching schemes which can be somewhat alleviated by the matching procedures which can apply lesser weight to the match at template boundaries than at frames within the word. A fourth problem is that word durations in the string are often grossly different (shorter) than the durations of the corresponding reference patterns. To alleviate this problem, one can use some time prenormalization procedure [21] to warp the word durations accordingly, or rely on reference patterns extracted from embedded word strings, as will be described later in this paper. Finally, the last problem associated with matching word strings is that the combinatorics of matching strings exhaustively (i.e., by trying all combinations of reference patterns in a sequential manner) is prohibitive.

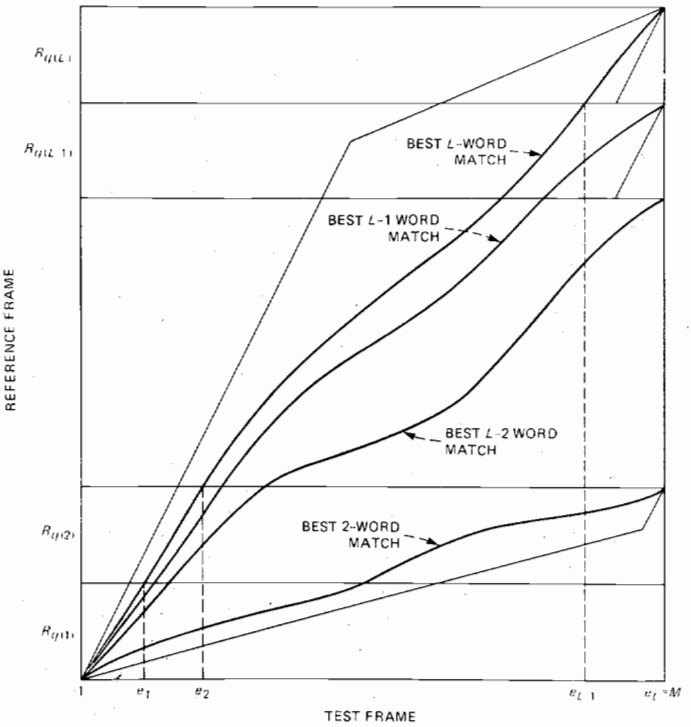


Fig. 2. Sequence of level building DTW warps to provide best word sequences of several different lengths.

A number of different ways of solving the connected word recognition problem have been proposed which avoid the plague of combinatorics mentioned above. Among these algorithms are the 2-level DP approach of Sakoe [5], the level building approach of Myers and Rabiner [6], the parallel single stage approach of Bridle *et al.* [7], and the nonuniform sampling approach of Gauvain and Mariani [8]. Although each of these approaches differs greatly in implementation, all of them are similar in that the basic procedure for finding $\boldsymbol{R}^s$ is to solve a time-alignment problem between $\boldsymbol{T}$ and $\boldsymbol{R}^s$ using dynamic time warping (DTW) methods.

The level building DTW based approach to connected word recognition is illustrated in Fig. 2. Shown in this figure are the warping paths for all possible length matches to the test pattern, along with the implicit word boundary markers $(e_1, e_2, \cdots, e_{L-1}, e_L)$ for the dynamic path of the L-word match. The level building algorithm has the property that it builds up all possible L-word matches one level (word in the string) at a time. For each string match found, a segmentation of the test string into appropriate matching regions for each reference word in $\boldsymbol{R}^s$ is obtained. In addition, for every string length L, the best β matches (i.e., the β lowest distance L-word strings) can be found. The details of the actual level building algorithm are available elsewhere [6], and will not be discussed here. Instead we will rely on the properties of the algorithm, men-

tioned above, to show how we can use them to obtain improved speaker independent word reference patterns.

A. Obtaining Reference Patterns for Connected Word Recognition

The "standard" set of word reference patterns used in most connected word recognition systems is basically a set of isolated word patterns. For speaker trained systems, usually one robust isolated word reference pattern is obtained; for speaker independent systems of a set of Q patterns are used for each word, where the Q patterns are extracted from a clustering analysis of a large set of patterns (typically 100 or more) of each word spoken by different talkers.

For the digits vocabulary we have made two small modifications to the standard isolated word pattern reference set. First we have created two distinct sets of patterns for the digit /8/; one with a t release at the end of the word, one without a t release at the end of the word. Thus we have a good representation of 8's both within strings, and at the end of strings (where it is usually released). The second modification we have made is to artifically lengthen the patterns for the digits 2 and 8 (without the release) by inserting "silence frames" at the beginning of 2 (to simulate the initial stop) and at the end of 8 (to simulate the final unreleased stop). For the digit 2 a silence duration of 45 ms was used; for the digit 8 a silence duration of 90 ms was used (the silence durations were optimized in a small pilot test).

To the standard isolated digit reference set, two sets of embedded digit patterns were added. These sets denoted as the noncoarticulated (NC) set, and the coarticulated (CO) set, were obtained as follows. For each of 80 talkers (40 men, 40 women) a speaker trained robust set [12] of isolated digit tokens was obtained. Then the embedded training algorithm [11] was used on 3-digit sequences to extract robust embedded versions of each of the digits in both NC and CO positions as follows. We denote the spoken 3-digit test sequence (either NC or CO) as $\boldsymbol{T}$. Since the spoken digits in the 3-digit test string are known, a reference pattern $\boldsymbol{R}$ is constructed by concatenating the 3 individual isolated digit patterns. A DTW alignment between $\boldsymbol{T}$ and $\boldsymbol{R}$ is then performed, thereby providing a segmentation of $\boldsymbol{T}$ into the three regions corresponding to each spoken digit. The embedded digit was extracted as the middle of the three digits; the initial and final digits were not used to give training patterns, since the influence of context on these digits was much smaller than for the middle digit. The embedded digit pattern, saved in a temporary word store, was compared to all previous occurrences of that digit in the store again using a DTW alignment procedure. For each such comparison, a distance score was obtained. If any distance score fell below a specified threshold, then the pair of digits giving the minimum distance among all versions in the store were averaged, after time alignment, and the resulting (embedded) reference pattern was saved in a permanent store. This procedure was iterated until an embedded reference pattern was obtained for each of the digits.

The above embedded training algorithm was used to extract NC and CO reference patterns for each digit. The set of sequences used for the embedded training is given in Table I. Each talker was requested to speak at a "normal" rate. Because of the high degree of variability of the embedded digits it often took 5 or more training sequences to extract a reliable embedded digit token, especially for the CO sequence. Hence for each talker it took between 10 and 30 min to obtain a single set of embedded digit tokens (either NC or CO tokens). For the 80 talker population, recording of training data took about three months with 1-2 h of recording per day.

TABLE I
TRAINING SEQUENCES FOR EMBEDDED DIGITS

NC Sequences			CO Sequences		
614	615	616	919	119	019
123	725	327	020	729	421
234	436	633	438	638	738
346	343	645	341	648	349
256	253	155	651	458	251
261	168	569	866	663	766
173	475	577	671	678	672
668	468	768	668	468	768
693	695	697	191	991	198
104	106	103	601	708	009
681	688	689	388	387	389
617	613	617	011	918	118
624	526	426	921	628	529
335	537	737	531	831	839
247	544	746	549	741	548
357	454	655	759	350	059
369	461	768	667	664	867
974	376	274	679	670	776
568	368	168	568	368	168
694	696	693	199	999	998
903	907	906	309	101	408
689	681	688	386	381	383

A set of speaker independent embedded digit tokens was obtained by clustering the 80 versions of each robust embedded digit using standard clustering techniques. Because of the high variability of the training data, the degree to which the embedded tokens clustered was significantly smaller than the degree to which the isolated tokens clustered; hence, on average only about six reliable clusters (with three or more training tokens) were found for each embedded digit. Thus, a typical speaker independent digits set consisted of 12 isolated tokens, six NC tokens, and six CO tokens for each digit or a total of $24 \times 11 = 264$ reference patterns (recall that the digit 8 was recorded both with and without t releases; hence, there were 11 digits).

III. EXPERIMENTAL EVALUATION OF SPEAKER INDEPENDENT CONNECTED DIGIT RECOGNITION

To test the effectiveness of the embedded digit training data in a speaker independent connected digit recognition mode, an evaluation was carried out in which each of 19 talkers (9 male, 10 female) spoke 40 randomly generated digit strings at both a deliberate (i.e., carefully articulated digits) and a normal talking rate. The 40 digit strings varied in length from 2 to 5 digits (with equal proportions of each string length); hence the average string length was 3.5 digits. No restrictions on digits within the string were used; hence, multiple digits often occurred within strings. The set of 40 digit strings was different for each talker. All recordings were made over a standard dialed-up telephone line, with a different line used for each talker. The 19 test talkers were not used in the training set of 80 talkers from which the speaker independent digit reference patterns were obtained.

A series of preliminary recognition tests were run to tune the parameters of the level building connected digit recognizer.

It was found that the optimum values of the level building parameters were essentially identical to those used in earlier digit recognition tests. The optimum parameter values obtained in this preliminary evaluation were as follows.

ϵ = width of DTW search region = 15 frames
M_T = multiplier for interlevel scores = 1.4
δ_{END} = search region at end of string = 4 frames
δ_{R1} = number of frames to skip at beginning of template = 0
δ_{R2} = number of frames to skip at end of template, variable.

The nominal choice for δ_{R2} was 4 frames for all isolated templates except 8 (both versions) and 6 for which δ_{R2} was 0. For all embedded templates (which were naturally reduced in length) the optimum choice of δ_{R2} was 0; i.e., no frames could be skipped at the end of the embedded references while matching the test sequences.

Using these optimum values of the level building parameters, a series of four recognition tests was run. In the first test only the standard isolated digit templates (12/digit) were used. This experiment provided a baseline comparison for measuring improvements in performance due to using embedded digit templates. For the last 3 tests a total of 24 templates per digit were used, i.e., 12 isolated, plus 6 NC embedded, plus 6 CO embedded. (Recall that we found that only six reliable embedded digit clusters could be obtained for the 80 talker training database.) The differences in the last three tests involved the decision rule; in particular the *K*-nearest neighbor (KNN) rule was used with values of KNN = 1, 2, and 3, respectively.

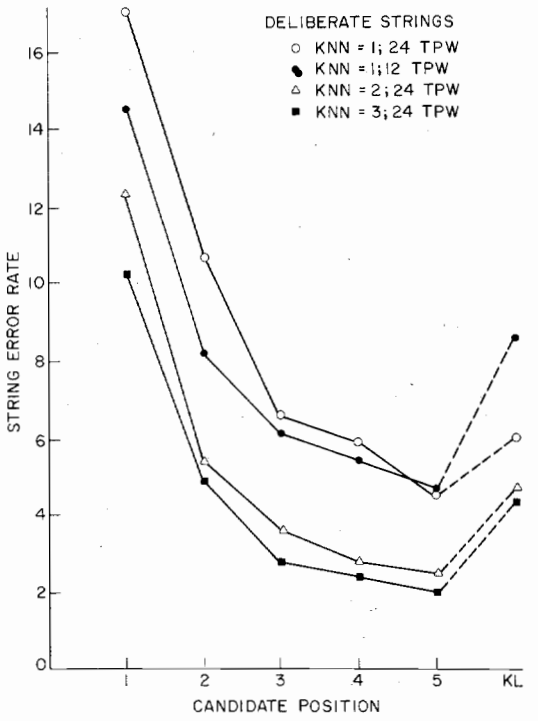
The results of the recognition evaluation tests are given in Figs. 3-5, and in Tables II and III. Table II gives the average error rates (across all 19 talkers) for each of the four recognition tests as a function of the position of the correct string in the ordered list of best strings. Hence, position 1 means the correct string had the lowest distance over all possible strings of any length (number of digits). Position 2 means the correct string had the second lowest distance over all strings, etc. Position KL (known length) means the correct string had the lowest distance among digit strings with the known correct number of digits.

Results are given in Table II for both deliberately spoken test strings (a) and for naturally spoken test strings (b). The results of Table II are shown plotted in Figs. 3(a) and 3(b). Based on Table II and Fig. 3, the following observations can be made.

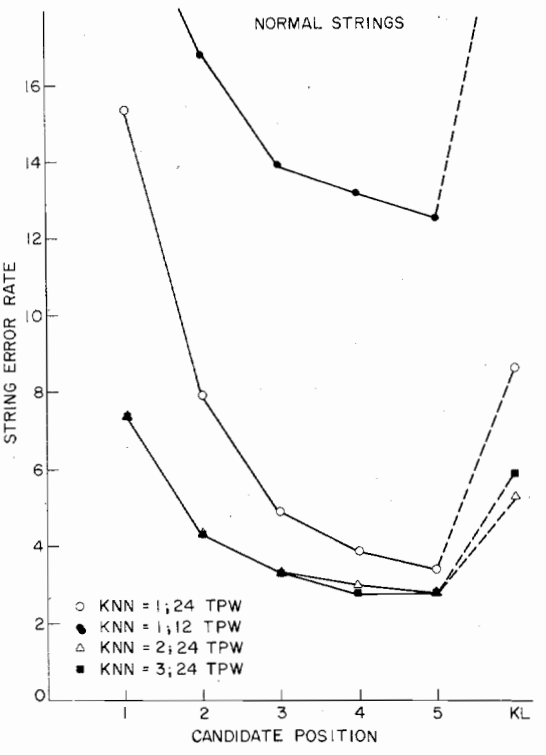
1) For deliberately spoken digits (with KNN = 1), addition of embedded training patterns tends to *decrease* the performance of the recognizer for strings of unknown length. However, for known length strings, a clear improvement in performance is obtained using the reference set with the embedded digit tokens.

2) The use of the KNN rule tends to greatly improve the performance of the recognizer for deliberately spoken strings. This effect is strongly seen in the results using KNN = 2 versus those for KNN = 1. The improvement in performance for KNN = 3 is marginal over that obtained for KNN = 2.

3) For normally spoken strings (with KNN = 1), a dramatic improvement in performance is obtained using the embedded training patterns. In position 1 the error rate falls from 25.3 percent for 12 isolated training patterns to 15.3 percent for 24 isolated plus embedded training patterns for digit. For the known length case, the error rate falls from 22 percent to 8.6 percent.



(a)



(b).

Fig. 3. String error rate versus candidate position for the four recognition tests for (a) deliberate strings, and (b) normal rate strings. The individual symbols denote measured error rates for the different KNN rules, and for the specified number of templates per word (TPW).

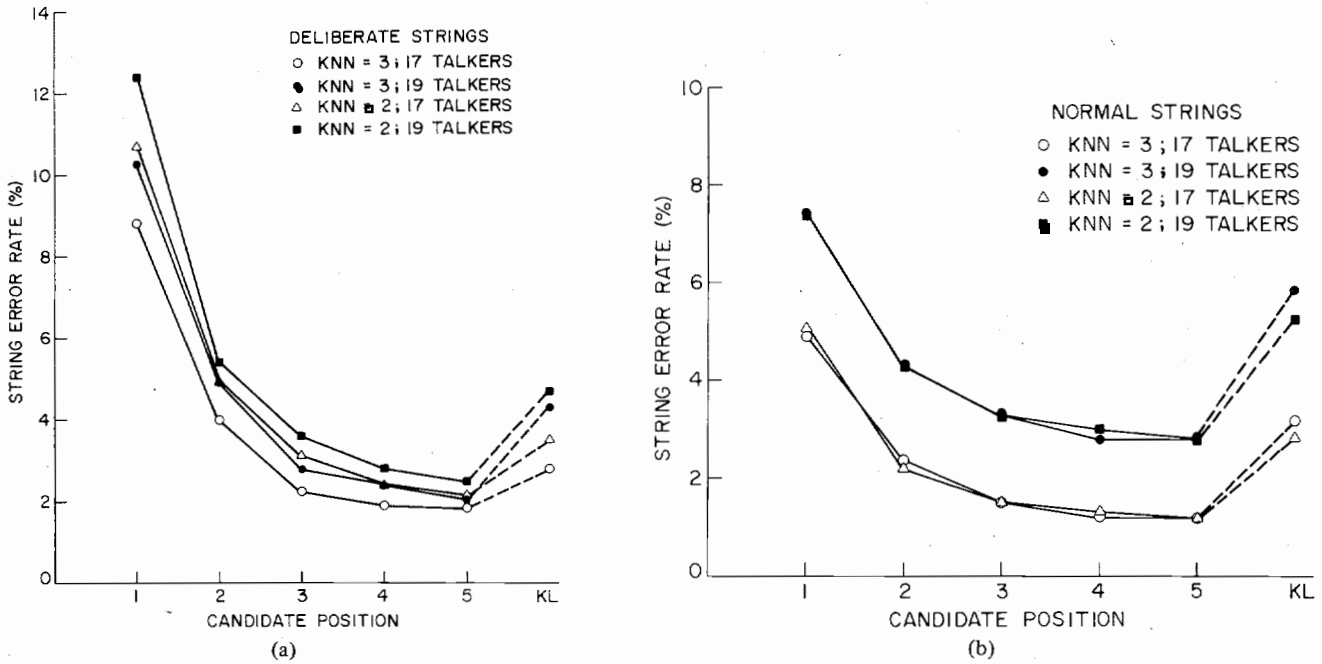


Fig. 4. String error rate versus candidate position for a 19- and a 17-talker database and for KNN = 2 and 3 for (a) deliberate strings and (b) normal rate strings.

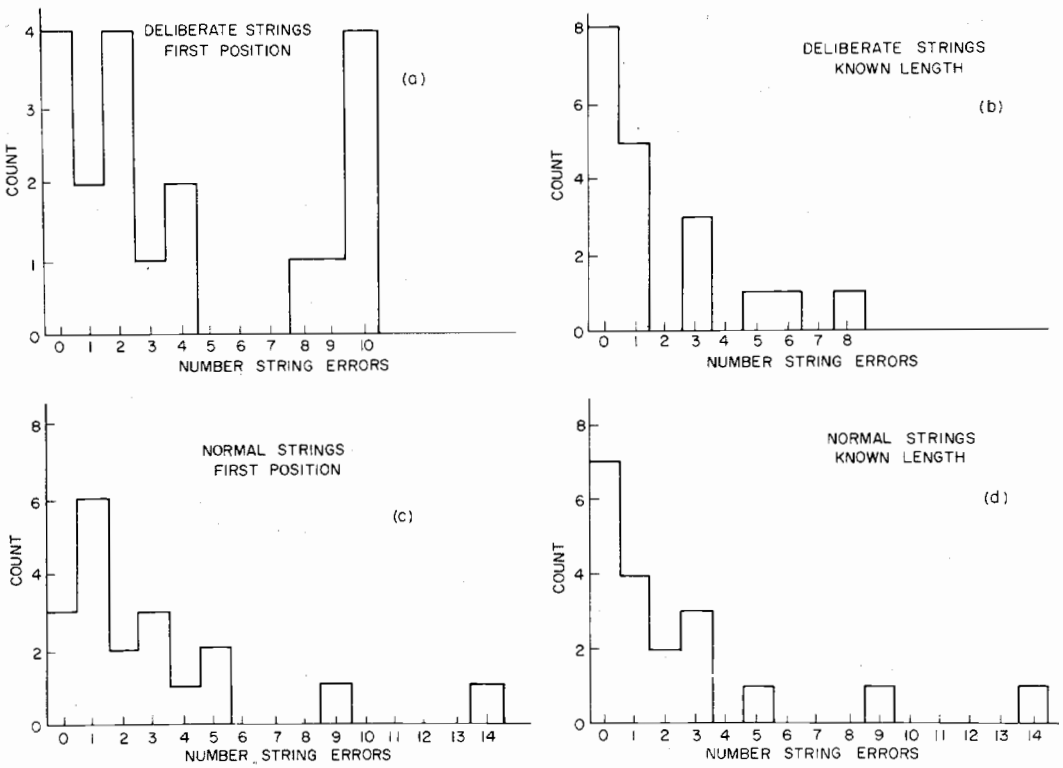


Fig. 5. Histograms of the number of string errors for [(a) and (b)] deliberate, and [(c) and (d)] normal rate strings. Parts (a) and (c) are first position counts and parts (b) and (d) are known string length counts.

4) The use of the KNN rule again lends to good improvements in performance of the recognizer for naturally spoken strings; however, differences in performance between KNN = 2 and KNN = 3 are statistically insignificant.

Since the best performance of the recognizer was obtained in the tests using 24 templates per digit with KNN = 3, Table III and Figs. 4 and 5 give further analysis of the results for this case. Table III shows individual talker error rates as a function of string position for both the deliberate (Table IIIa) and naturally spoken strings (Table IIIb). Also given in the tables are average string error rates for all 19 talkers, and for the best 18 and 17 talkers. It can be seen in Table III that 2 or 3 of the 19 talkers had error rates far worse than the average; hence, these talkers had a strong influence on the overall performance

TABLE II
RESULTS FOR FOUR CONNECTED-DIGIT RECOGNITION TESTS

Test Number	Number Templates per Digit	KNN Rule	Error Rate (%) in Position					
			1	2	3	4	5	KL
1	12*	1	14.6	8.2	6.1	5.4	4.7	8.6
2	24	1	17.1	10.7	6.6	5.9	4.5	6.0
3	24	2	12.4	5.4	3.6	2.8	2.5	4.7
4	24	3	10.3	4.9	2.8	2.4	2.0	4.3

(a) Error Rates for Deliberately Spoken Digit Strings

Test Number	Number Templates per Digit	KNN Rule	Error Rate (%) in Position					
			1	2	3	4	5	KL
1	12*	1	25.3	16.8	13.9	13.2	12.5	22.0
2	24	1	15.3	7.9	4.9	3.9	3.4	8.6
3	24	2	7.4	4.3	3.3	3.0	2.8	5.3
4	24	3	7.4	4.3	3.3	2.8	2.8	5.9

(b) Error Rates for Normally Spoken Digit Strings

* Templates derived from isolated digits.

TABLE III
INDIVIDUAL TALKER RESULTS FOR SPEAKER INDEPENDENT CONNECTED DIGIT RECONSTRUCTION

Talker Number	Error Rate (%) in Position					
	1	2	3	4	5	KL
1	0	0	0	0	0	0
2	5	0	0	0	0	2.5
3	5	2.5	0	0	0	2.5
4	0	0	0	0	0	0
5	5	2.5	0	0	0	0
6	10	0	0	0	0	0
7	5	2.5	0	0	0	2.5
8	2.5	0	0	0	0	0
9	7.5	5.0	0	0	0	2.5
10	25	15	7.5	7.5	7.5	7.5
11	25	7.5	7.5	5.0	5.0	12.5
12	20	7.5	5.0	5.0	0	20
13	22.5	12.5	12.5	12.5	10	7.5
14	0	0	0	0	0	0
15	25	17.5	10	7.5	7.5	15
16	0	0	0	0	0	0
17	25	12.5	7.5	7.5	7.5	7.5
18	2.5	0	0	0	0	0
19	10	7.5	2.5	0	0	2.5
Average (19)	10.3	4.9	2.8	2.4	2.0	4.3
Average (18)	9.4	4.2	2.4	2.1	1.7	3.8
Average (17)	8.8	4.0	2.2	1.9	1.8	2.8

(a) Error Rates for Deliberately Spoken Digit Strings, KNN=3

Talker Number	Error Rate (%) in Position					
	1	2	3	4	5	KL
1	2.5	0	0	0	0	0
2	2.5	0	0	0	0	0
3	7.5	2.5	2.5	2.5	2.5	7.5
4	2.5	0	0	0	0	2.5
5	0	0	0	0	0	0
6	0	0	0	0	0	0
7	5	5	2.5	2.5	2.5	5
8	2.5	2.5	2.5	2.5	2.5	2.5
9	12.5	7.5	5	2.5	2.5	7.5
10	10	2.5	2.5	0	0	7.5
11	2.5	2.5	2.5	2.5	2.5	2.5
12	22.5	17.5	12.5	7.5	7.5	22.5
13	12.5	10	5	5	5	12.5
14	0	0	0	0	0	0
15	35	25	25	25	25	35
16	7.5	0	0	0	0	0
17	2.5	2.5	2.5	2.5	2.5	2.5
18	5	2.5	0	0	0	0
19	7.5	2.5	0	0	0	5
Average (19)	7.4	4.3	3.3	2.8	2.8	5.9
Average (18)	5.8	3.2	2.1	1.5	1.5	4.3
Average (17)	4.9	2.4	1.5	1.2	1.2	3.2

(b) Error Rates for Naturally Spoken Digit Strings, KNN=3

scores. To show the effects of eliminating these talkers on performance, the average error rates without talkers 15 and 12 were computed. Fig. 4 shows plots of the average error rate scores as a function of position in the list of candidate strings for the 19 and 17 talker cases, for both the KNN = 2 and KNN = 3 tests.

Fig. 5 shows histograms of the number of strings errors across the 19 talkers in both first position [(a) and (c)] and for known length strings [(b) and (d)], for both deliberate strings [(a) and (b)], and normal strings [(c) and (d)]. It can be seen in Table III and in Fig. 5 that the majority of talkers had excellent performance (the median string error rate was 2.5 percent in known length strings with almost half the talkers having no string errors for either deliberate or natural rate strings).

To understand the types of error made in connected digit recognition, a detailed analysis was made of the errors in recognition test 4, with 24 templates per digit using KNN = 3. The recognition errors were classified as being very close (VC), where the difference in distance between the correct string and the chosen string was less than 0.005, close (C), where the difference in distance between the correct string and the chosen string was less than 0.03 but greater than 0.005, and far (F) if the difference between the correct string and the chosen string was greater than 0.03. For the deliberate strings there were 13 VC, 42 C, and 22 F. For the normal strings there were 9 VC, 21 C, and 26 F. A further analysis of the deliberate string errors showed the following.

1) The 13 VC errors comprised 7 digit insertion errors (4 of these were the digit 8), and 6 digit substitution error (no regular pattern).

2) The 42 C errors comprised 31 digit insertion errors (10 of these were the digit 8, 10 were the digit 1, four were the digit 2, three were the digit 6, three were the digit 5, one was the digit 7), and 11 substitution errors (the only regular substitution was 9 to 1 which occurred four times for one talker).

Hence, the worst problem for deliberate strings was digit insertions (38 of 55 cases), generally involving short word reference patterns (e.g., 8, 1, 2). This problem is a consequence of using the embedded word patterns which often are very short duration. Thus, for the deliberate strings the known length case led to significantly better performance than the unknown length case.

For the normal rate digit strings, an analysis of the 9 VC and 21 C errors showed the following.

1) The 9 VC errors involved seven digit substitutions, one digit insertion, and one digit deletion. There was no regularity to the digit substitution errors.

2) The 21 C errors involved 15 digit substitutions, five digit insertions, and one digit deletion. Four of the digit substitutions were for one talker and involved using a 1 in place of a 9; all other such errors had no pattern. The digit insertions involved the digits 2 and 8 (two each). For normal rate strings,

TABLE IV
AVERAGE STRING ERROR RATES AS A FUNCTION OF THE NUMBER OF DIGITS IN THE STRING

	Number of Digits in String			
Type of Articulation	2	3	4	5
Deliberate	12.6	13.1	15.8	7.9
Normal	5.3	7.4	8.9	7.9

(a) String Error Rates (%) for Unknown Length Strings of Test 3 as a Function of the Number of Digits in the String

	Number of Digits in String			
Type of Articulation	2	3	4	5
Deliberate	2.1	3.2	4.2	7.9
Normal	2.6	4.2	6.3	7.9

(b) String Error Rates (%) for Known Length Strings of Test 3 as a Function of the Number of Digits in the String

only the shortest digit patterns have a chance of being inserted in a string.

Our overall analysis of the digit string errors indicated that digit insertions were the worst problem and occurred primarily for deliberately spoken strings. There seemed to be little pattern or regularity to the errors in the digit substitution cases for either deliberate or natural rate strings.

An additional analysis was made of the correlation between string length and string error rate for recognition test 3. The reults of this analysis are shown in Table IV which gives average string error rates (%) as a function of whether the string length was unknown (Table IVa) or known (Table IVb), and as a function of the number of digits in the string. For the deliberate strings of unknown length, the average string error rate for 5 digit strings is significantly lower than for strings of 2, 3, or 4 digits. This is because there is no digit insertion problem with 5-digit strings, and since digit insertions are the most severe problem with deliberately spoken digit strings, the string error rate for 5-digit strings appears to be the lowest. This fact is verified in the results of Table IVb, which shows that for known length strings, the average string error rate for 5-digit strings is significantly higher than for strings 2, 3, or 4 digits. In fact, we see a tendency for higher error rates as the number of digits in the string increases.

For normal rate strings we see a similar affect to that noted above, in that for unknown length strings, there is almost the same average string error rate for all string lengths; however, for known length strings, the error rate clearly increases as the number of digits in the string increases. It is also interesting to note that the average string error rates for both deliberate and normal strings are comparable for the known length strings for all string lengths. Hence, the use of embedded digit training words tends to compensate for the higher rates of articulation uniformly well across all length digit strings.

IV. DISCUSSION

The results presented in Tables II and III and shown in Figs. 3–5 have demonstrated the following.

1) The inclusion of embedded digit training improves string recognition accuracy significantly for normal rate digit strings (both known and unknown lengths), but lowers string accuracy for deliberate rate strings of unknown length, and raises string accuracy for deliberate rate strings of known length.

2) The use of higher values of KNN (2 and 3) leads to significant improvements in performance for both deliberate and normal rate digit strings.

3) For known length strings, average string error rates of about 5 percent can be maintained for both deliberate and normal rate strings.

4) For unknown length strings, average string error rates are from 1.5 to 2 times higher than in the case of known length strings. The increased error rate is due primarily to the ease of inserting short reference digits (8, 2, 1) in matching digit strings. The effect is more pronounced for deliberate rate strings than for normal rate strings.

5) For speaker independent recognition, most talkers were able to be recognized with fairly small string error rates (2.5 percent or below). However, there were a couple of talkers in the 19 tested whose error rates were from 5 to 10 times higher than the average of the other talkers. Informal listening did not indicate any obvious problem in the speech of such talkers; hence, we have no good explanation as to why these talkers were recognized so poorly. Our only possible explanation is that in an earlier study [11], it was found that even for speaker trained connected digit recognition, there were talkers whose string error rate was on the order of 50 percent. Hence, there is some element of digit variability in some talkers that appears to be poorly modeled in our current recognizer.

The question that remains is what can be done to improve performance on the task of speaker independent, connected digit recognition. There are at least three things that come to mind to help increase average digit string accuracy. The first is to improve the training by including far more talkers in the training data base. This path is one which must be taken before one can hope to use such a recognizer for a real-world application. However, in the laboratory, the cost of a significant increase in training data is as yet excessive (both manpower and processing) and will not be attempted in the immediate future. A second step to improve connected digit accuracy is to incorporate some form of task syntax into the problem so as to be able to automatically detect and correct string errors. For example, in the use of connected digit recognition for dialing a valid telephone number (i.e., 3 digit area code plus 7 digit telephone number), we can use the following syntactic information.

1) We expect only 10 digits in the set of strings spoken (typically a 3-digit area code followed by a pause, then a 3-digit exchange, then a pause, then a 4-digit number). Hence, we can handle the digit insertion problem by counting and detecting too many digits in one or more strings.

2) Of the $10^3 = 1000$ possible area codes, only 117 are valid in the U.S. Hence, we can check the first 3 recognized digits against the list of valid area codes and detect and possibly correct area code errors automatically.

3) Within a given area code there are $10^3 = 1000$ possible

exchanges. On average, there are about 200–400 exchanges within a given area code. Hence, one can apply simple table driven syntactic rules to detect and possibly correct exchange errors.

4) Finally, within a given area code and exchange there are $10^4 = 10\,000$ possible numbers. However, there are, in general, anywhere from 1 percent and 50 percent of the numbers which are unassigned or unused. Again, it is theoretically possible to check the actual number and see if it is valid, and therefore detect any possible correct number errors.

Other digit string entry systems have similar forms of syntactic constraints which can be utilized to improve connected digit recognition performance.

The third way to improve performance on the connected digit recognition task is to incorporate some side information into the recognizer. For example, it has been shown that the addition of an energy contour can halve the error rate in an isolated word recognizer for a vocabulary of 129 airline terms [14]. The use of energy information would possibly be of much more value in a connected digit recognizer than in the isolated word recognizer, because it would tend to eliminate the digit insertions which would have very poor energy matches to the test strings. Another technique which could be combined with the LPC based DTW recognizer is a hidden Markov model (HMM) recognizer [15]. Based on previous word with isolated digits it was shown that a high degree of disjointness existed between the errors made by an HMM recognizer, and those made on a conventional LPC recognizer. By combining the two isolated word recognizers, it was argued that a recognizer could be obtained with virtually no errors. One would expect that such properties could also be explained in the connected digit recognition problem.

V. Summary

We have shown how an improved digit training technique developed previously for speaker trained, connected word recognizers, could be combined with a standard token clustering analysis to give an improved speaker independent connected digit recognition system. Further we have shown that significant improvements in performance are obtained by using the K-nearest neighbor rule with values of KNN = 2 or 3. The results of an evaluation test with 19 talkers indicate that average string accuracies of about 95 percent can be obtained for both deliberate and normal talking rates if the number of digits in the string is known.

References

[1] T. B. Martin, "Practical applications of voice input to machines," *Proc. IEEE*, vol. 64, pp. 487–501, Apr. 1976.

[2] S. Moshier, "Talker independent speech recognition in commercial environments," in *Speech Commun. Papers 97th ASA Meeting*, June 1979, pp. 551–553.

[3] F. Itakura, "Minimum prediction residual principal applied to speech recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-23, pp. 67–72, Feb. 1975.

[4] G. R. Doddington and T. B. Schalk, "Speech recognition: turning theory to practice," *IEEE Spectrum*, vol. 18, pp. 26–32, Sept. 1981.

[5] H. Sakoe, "Two level Dp-matching–A dynamic programming based pattern matching algorithm for connected word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-27, pp. 588–595, Dec. 1979.

[6] C. S. Myers and L. R. Rabiner, "Connected digit recognition using a level building DTW algorithm," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, pp. 351–363, June 1981.

[7] J. S. Bridle, M. D. Brown, and R. M. Chamberlain, "An algorithm for connected word recognition," *Automatic Speech Analysis and Recognition*, J. P. Haton, Ed., 1982, pp. 191–204.

[8] J. L. Gauvain and J. Mariani, "A method for connected word recognition and word spotting on a microprocessor," in *Proc. ICASSP*, May 1982, pp. 891–894.

[9] J. Peckham, J. Green, J. Canning, and P. Stephens, "LOGOS–A real time hardware continuous speech recognition system," in *Proc. ICASSP*, May 1982, pp. 863–866.

[10] S. E. Levinson and K. L. Shipley, "A conversational mode airline information and reservation system using speech input and output," *Bell Syst. Tech. J.*, vol. 59, pp. 119–137, Jan. 1980.

[11] L. R. Rabiner, A. Bergh, and J. G. Wilpon, "An improved training procedure for connected-digit recognition," *Bell Syst. Tech. J.*, vol. 61, pp. 981–1001, July–Aug. 1982.

[12] M. H. Kuhn and H. H. Tomaschewski, "Improvements in isolated word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-31, pp. 157–167, Feb. 1983.

[13] L. R. Rabiner and J. G. Wilpon, "A simplified, robust training procedure for speaker trained, isolated word recognition systems," *J. Acoust. Soc. Amer.*, vol. 68, pp. 1271–1276, Nov. 1980.

[14] M. K. Brown and L. R. Rabiner, "On the use of energy in LPC-based recognition of isolated words," *Bell Syst. Tech. J.*, vol. 61, pp. 2971–2987, Dec. 1982.

[15] L. R. Rabiner, S. E. Levinson, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker independent, isolated word recognition," *Bell Syst. Tech. J.*, vol. 62, pp. 1075–1105, Apr. 1983.

Lawrence R. Rabiner (S'62–M'67–SM'75–F'75) was born in Brooklyn, NY, on September 28, 1943. He received the S.B. and S.M. degrees simultaneously in June 1964, and the Ph.D. degree in electrical engineering in June 1967, all from the Massachusetts Institute of Technology, Cambridge.

From 1962 through 1964 he participated in the cooperative plan in electrical engineering at ATT, Bell Laboratories, Whippany and Murray Hill, NJ. He worked on digital circuitry, military communications problems, and problems in binaural hearing. Presently he is engaged in research on speech recognition and digital signal processing techniques at Bell Laboratories, Murray Hill. He is coauthor of the books *Theory and Application of Digital Signal Processing* (Englewood Cliffs, NJ: Prentice-Hall, 1975), *Digital Processing of Speech Signals* (Prentice-Hall, 1978), and *Multirate Digital Signal Processing* (Prentice-Hall, 1983).

Dr. Rabiner is a member of Eta Kappa Nu, Sigma Xi, Tau Beta Pi, and a Fellow of the Acoustical Society of America. He is also a member of the National Academy of Engineering.

Jay G. Wilpon was born in Newark, NJ on February 28, 1955. He received the B.S. and A.B. degrees (cum laude) in mathematics and economics, respectively, from Lafayette College, Easton, PA in 1977, and the M.S. degree in electrical engineering/computer science from Stevens Institute of Technology, Hoboken, NJ, in 1982.

He is currently a Member of the Technical Staff with the Acoustics Research Department, ATT, Bell Laboratories, Murray Hill, NJ. Since 1977 he has been engaged in speech communications research, concentrating on problems of speech recognition. He has designed several isolated and connected speech recognition systems and has published extensively in this area. His interests include basic research in the field of speech recognition, training algorithms for speaker dependent and speaker independent recognizers, and speech endpoint detection.

Ann M. Quinn was born in Plainfield, NJ, on July 31, 1951. She graduated from Rutgers University, New Brunswick, NJ, with a degree in linguistics and speech science.

In 1969 she joined ATT Bell Laboratories, Murray Hill, NJ, and is presently working in the Acoustics Research Department.

Sandra G. Terrace was born in The Bronx, NY on April 22, 1953. She received the B.S. degree (summa cum laude) in mathematics from Northeastern University, Boston, MA, in 1975.

She works primarily from her home in Chelmsford, MA, where she and her husband own and operate their own computer software consulting firm. She is also a computer software consultant to the Acoustics Research Department, ATT, Bell Laboratories, Murray Hill, NJ. Since April 1980 she has been engaged in the development of speech recognition and processing algorithms with the CSPI MAP array processor. Programs have been developed using MAP Assembly languages and Data General Fortran to accomplish isolated and connected speech recognition, speech synthesis, and speech detection.

Time Delay Estimation by Generalized Cross Correlation Methods

MORDECHAI AZARIA AND DAVID HERTZ

Abstract—The problem of estimating time delay by cross correlation methods is reexamined for the whole class of stationary signals.

Expressions are derived for the estimation mean square error (MSE) by the cross correlation method, and are shown to be identical to previously published results for Gaussian signals.

The generalized cross correlation method is also analyzed, and the optimal weight function for this method is derived. It is shown to be identical to that derived for Gaussian signals by the maximum likelihood method.

For the cross correlation method a simplified MSE expression is derived, which is to be used instead of a previously published result.

I. Introduction

THE problem of estimating time delay of arrival (TDOA) is reexamined for the whole class of stationary signals. Both cross correlation and generalized cross correlation methods are analyzed [5], [9], [10]. This problem has been the theme of a recent special issue [1], and numerous papers were published thereafter. Assume the model

$$x(t) = s(t) + n_1(t)$$
$$y(t) = s(t - D) + n_2(t) \qquad -T/2 \leqslant t \leqslant T/2 \tag{1}$$

where the signal $s(t)$ and noise $n_1(t)$, $n_2(t)$ are real baseband signals. D is the unknown delay, T is the observation time, and the following assumptions hold.

Assumption a: The signal $s(t)$ and noise $n_1(t)$ and $n_2(t)$ are stationary band limited zero mean signals uncorrelated with each other.

Assumption b: The correlation durations of the signals $s(t)$, $n_1(t)$ and $n_2(t)$: $|D| + \tau_s$, τ_{n_1}, τ_{n_2}, respectively, are very small compared to the observation time T. Ergodicity of the signal $s(t)$ is also assumed.

In Section II, an error analysis of TDOA estimation by the cross correlation method is carried out.

The delay D can be estimated by $\tau = \hat{D}$ for which the cross correlation function $\phi(\tau)$ is maximized, i.e.,

$$\phi(\hat{D}) = \max_{\tau} \phi(\tau) = \max_{\tau} \int_{-T/2}^{T/2} dt\, x(t)\, y(t + \tau). \tag{2}$$

In this paper, it is shown that the estimator $\hat{D}$ of D is unbiased and an expression for its mean square error (MSE), as a function of the observation time T and the autospectra of the signal $s(t)$ and the noise $n_1(t)$ and $n_2(t)$, is derived. In the derivations it is assumed that TDOA estimation error is very small compared to the correlation duration τ_s of $s(t)$, i.e., $|\hat{D} - D| \ll \tau_s$.

Therefore in $\hat{D}$'s domain, the autocorrelation function $R_{ss}(\hat{D} - D)$ of $s(t)$ can be approximated by a parabola. As will be shown, the validity of this approximation depends on signal to noise ratios (SNR), observation time T and signal and noise bandwidths.

The expression for the MSE is identical to the result obtained in [8] assuming Gaussian signals, and to the leading term of the result obtained in [2] assuming Gaussian signals and a Gaussian autocorrelation function.

Manuscript received April 19, 1983; revised October 24, 1983.

M. Azaria is with Rafael Israel, Haifa, 31021, Israel.

D. Hertz is with the Technion–Israel Institute of Technology, Haifa, 32000 Israel.

0096-3518/84/0400-0280 $01.00