THE BELL SYSTEM TECHNICAL JOURNAL
Vol. 60, No. 5, May–June 1981
Printed in U.S.A.

A Two-Pass Pattern-Recognition Approach to Isolated Word Recognition

By L. R. RABINER and J. G. WILPON

(Manuscript received December 15, 1980)

One of the major drawbacks of the standard pattern-recognition approach to isolated word recognition is that poor performance is generally achieved for word vocabularies with acoustically similar words. This poor performance is related to the pattern similarity (distance) algorithms that are generally used in which a global distance between the test pattern and each reference pattern is computed. Since acoustically similar words are, by definition, globally similar, it is difficult to reliably discriminate such words, and a high error rate is obtained. By modifying the pattern-similarity algorithm so that the recognition decision is made in two passes, we can achieve improvements in discriminability among similar words. In particular, on the first pass the recognizer provides a set of global distance scores which are used to decide a class (or a set of possible classes) in which the spoken word is estimated to belong. On the second pass we use a locally weighted distance to provide optimal separation among words in the chosen class (or classes), and make the recognition decision on the basis of these local distance scores. For a highly complex vocabulary (letters of the alphabet, digits, and three command words), we obtain recognition improvements of from 3 to 7 percent using the two-pass recognition strategy.

I. INTRODUCTION

As illustrated in Fig. 1, the "standard" pattern recognition approach to isolated word recognition is a three-step method consisting of feature measurement, pattern similarity determination, and a decision rule for choosing recognition candidates. This pattern recognition model has been applied to a wide variety of word recognition systems with great success.[1–8] However, the simple, straightforward approach to word recognition, shown in Fig. 1, runs into difficulties for complex vocabularies, i.e., vocabularies with phonetically similar words. For

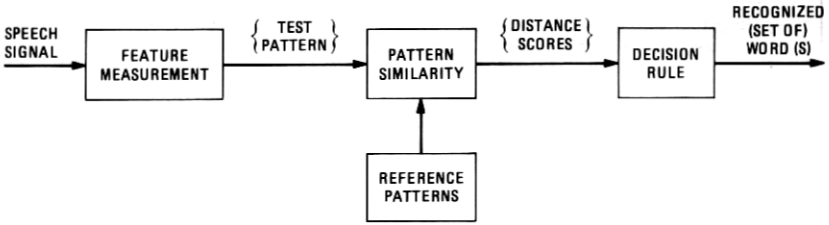


Fig. 1—Block diagram of standard approach to isolated word recognition.

example, recognition of the vocabulary consisting of the letters of the alphabet would have problems with letters in the sets

$$\phi_1 = \{A, J, K\},$$
$$\phi_2 = \{B, C, D, E, G, P, V, T, Z\},$$
$$\phi_3 = \{Q, U\},$$
$$\phi_4 = \{I, Y\},$$
$$\phi_5 = \{L, M, N\},$$
$$\phi_6 = \{F, S, X\}.$$

Similarly, recognition of the computer terms of Gold[9] might lead to confusions among the set containing four, store, and core. In the above cases the problems are due to the inherent acoustic similarity (overlap) between sets of words in the vocabulary. It should be clear that this type of problem is essentially unrelated to vocabulary size (except when we approach very large vocabularies), since a large vocabulary may contain no similar words (e.g., the Japanese cities list of Itakura[4]), and a small vocabulary may contain many similar words (e.g., the letters of the alphabet).

The purpose of this paper is to propose, discuss, and evaluate a modified approach to isolated word recognition in which a two-pass method is used. The output of the first recognition pass is an ordered set of word classes in which the unknown spoken word is estimated to have occurred, and the output of the second pass is an ordered list of word candidates within each class obtained from the first pass. The computation for the first pass is similar in nature but often reduced in magnitude from that required for the standard one-pass word recognizer. The computation of the second pass consists of using an "optimally" determined word discriminator to separate words within the equivalence class. In Section II, we present the two-pass recognizer, and discuss its philosophy and method of implementation. In Section III, we give an evaluation of the effectiveness of the two-pass approach for a vocabulary consisting of the 26 letters of the alphabet, the 10 digits, and the command words STOP, ERROR, and REPEAT. Finally, in Section IV, we summarize the results and show how they are applicable to practical speech recognition systems.

II. THE TWO-PASS RECOGNIZER

Assume the word vocabulary consists of V words. The ith word, v_i, is represented by the word template $\mathbf{R}_i$, $i = 1, 2, \cdots, V$, where each $\mathbf{R}_i$ is a multidimensional feature vector. Similarly, we denote the test pattern as $\mathbf{T}$ (corresponding to the spoken word q in the vocabulary), where $\mathbf{T}$ is again a multidimensional feature vector. For simplicity we assume that the pattern similarity and distance computation is carried out using the "normalize and warp" procedure described by Myers et al.,[10] and illustrated in Fig. 2. A "standard" word duration of N frames is adopted, and each reference pattern is linearly warped to this duration. We call the warped reference patterns $\tilde{\mathbf{R}}_i$. Similarly, the test pattern is linearly warped to a duration of N frames, yielding the new pattern $\tilde{\mathbf{T}}$. A dynamic time-warping alignment algorithm then computes the "standard" distance

$$D(\tilde{\mathbf{T}}, \tilde{\mathbf{R}}_i) = \frac{1}{N} \sum_{k=1}^{N} d(\tilde{\mathbf{T}}(k), \tilde{\mathbf{R}}_i(w(k))), \tag{1}$$

where $d(\tilde{\mathbf{T}}(k), \tilde{\mathbf{R}}_i(l))$ is the local distance between frame k of the test pattern, and frame l of the ith reference pattern, and $w(k)$ is the time-alignment mapping between frame k of the test pattern, and frame $w(k)$ of the ith reference pattern. The total distance D of eq. (1) is only a function of i.

We define the local distance of the kth frame of the test pattern to the $w(k)$th frame of the ith reference pattern as $d_i(k)$, where

$$d_i(k) = d(\tilde{\mathbf{T}}(k), \tilde{\mathbf{R}}_i(w(k))), \tag{2}$$

so $D(\tilde{\mathbf{T}}, \tilde{\mathbf{R}}_i)$ of eq. (1) can be written as

$$D(\tilde{\mathbf{T}}, \tilde{\mathbf{R}}_i) = \frac{1}{N} \sum_{k=1}^{N} d_i(k). \tag{3}$$

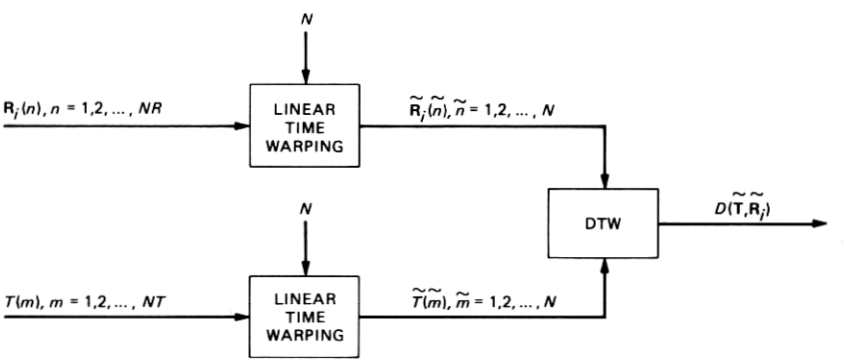


Fig. 2—Block diagram of the normalize-and-warp procedure for equalizing the lengths of words.

If $\bar{\mathbf{R}}_i$ corresponds to the correct reference for the spoken word $\bar{\mathbf{T}}$ (i.e., $i = q$), then we would theoretically expect the local distance $d_q(k)$ to be *independent* of k, with d assuming values from a χ^2 distribution with p (eight for the system we are using) degrees of freedom for the case where the speech features are those of an LPC model and the log likelihood distance measure is used for the local distance.[11,12] Thus, if we plotted $d_q(k)$ versus k, we would expect it to vary around some expected value $\hat{d}$ where

$$\hat{d} = E[d_q(k)] = E[\chi_p^2]. \tag{4}$$

An example of a typical curve of $d_q(k)$ versus k is given in Fig. 3a.

If we now examine the typical behavior of the curve of $d_i(k)$ versus k when $i \neq q$, we see that one of two types of behavior generally occurs. When word q is acoustically very different from word i, then $d_i(k)$ is generally large [compared to $\hat{d}$ of eq. (4)] for *all* values of k, and the overall distance score D of eq. (3) is large. This case is illustrated in Fig. 3b. However, when we have acoustically similar

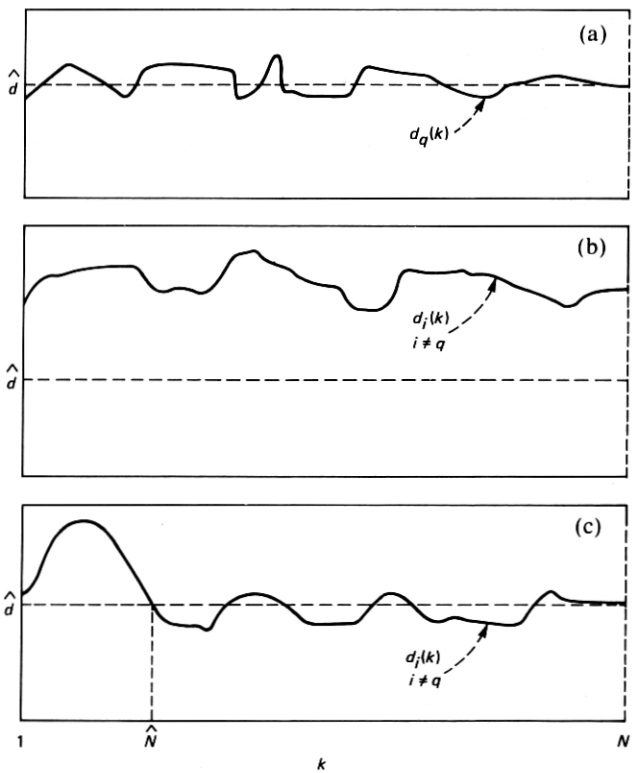


Fig. 3—Curves of $d_i(k)$ versus k for three cases.

words, then generally $d_i(k)$ will be approximately equal to $d_q(k)$ for all values of k in acoustically *identical* regions, and will be larger than $d_q(k)$ only in acoustically dissimilar regions. An example in which the dissimilar region occurs at the beginning of the word (the first $\hat{N}$ frames) is shown in Fig. 3c.

The key point to be noted from the above discussion is that when the vocabulary contains words that are acoustically similar, and one of these similar words is spoken (i.e., it is the test utterance), then the total distance scores for these similar words consists of a random component [because of the variations of $d(k)$ in the similar regions] and a deterministic difference (because of the differences in the dissimilar regions). In cases when the size of the dissimilar region is small (i.e., $\hat{N} \ll N$ in Fig. 3c), then the random component of the distance score can (and often does) outweigh the true difference component, causing a potential recognition error. For highly complex vocabularies (e.g., the letters of the alphabet), this situation occurs frequently.

One possible solution to the above problem would be to modify the overall distance computation so that more weight is given to some regions of the pattern than others. For example, we could consider a weighted overall distance of the form

$$D(\tilde{\mathbf{T}}, \tilde{\mathbf{R}}_i) = \frac{\sum_{k=1}^{N} W(k) d(\tilde{\mathbf{T}}(k), \tilde{\mathbf{R}}_i(w(k)))}{\sum_{k=1}^{N} W(k)}, \tag{5}$$

where $W(k)$ is an arbitrary frame weighting function, and the denominator of eq. (5) is used for distance normalization. The problem with eq. (5) is that a "good" weighting function is difficult to define since the "optimal" set of weights is clearly a function of the "actually" spoken word (q) and the reference pattern being used (i). Furthermore, any weighting that would help discriminate between acoustically similar words, would tend to hurt the discrimination between acoustically different words.

The above discussion suggests that a reasonable approach would be a two-pass recognition strategy in which the first pass would decide on an ordering of word "equivalence" classes (in which sets of acoustically similar words occurred), and the second pass would order the individual words within each equivalence class. For the first-pass recognition an unweighted (normal) distance would be used, and for the second pass a weighted distance would be used. In order to implement such a two-pass recognizer, a number of important questions must be answered, including:

(i) How do we "automatically" choose the word equivalence classes for each new vocabulary?

(*ii*) How do we determine class distance scores for the first recognition pass?

(*iii*) How do we determine weighting functions for the second recognition pass?

(*iv*) How do we generate weighted distance scores for the second recognition pass?

(*v*) How do we combine results from both recognition passes to give a final, overall set of distance scores and word orderings?

Some possible answers to each of these questions are given in the following sections.

2.1 Generation of word equivalence classes

Given the V vocabulary words $v_1, v_2, \cdots, v_V$, we would like to find a procedure for mapping words into acoustic equivalence classes ϕ_j, $j = 1, 2, \cdots, J$, where $J \leq V$. There are at least two reasonable approaches for solving this problem; one is a theoretical approach, the other an experimental one.

For the theoretical approach we can generate a "word-by-word" distance matrix D_w, on the basis of the phonetic transcriptions of the vocabulary entries. In order to do this we need to define a "phoneme" distance matrix, d_p, a distance cost for inserting a phoneme, d_I, and a distance cost for deleting a phoneme, d_D. The phoneme distance matrix could be a count of the number of distinctive features that have to be changed to convert from one phoneme to another.[13] A total word-by-word distance is then defined by a dynamic time-warp match between the words, with a vertical step representing an insertion, and a horizontal step representing deletion. Figure 4a illustrates this procedure for the words eight and *J*, and Figure 4b for the words one and nine. For the words eight and *J*, the optimum path is an insertion (of *J*), match between e^I and e^I, and a deletion of *t*, giving a distance

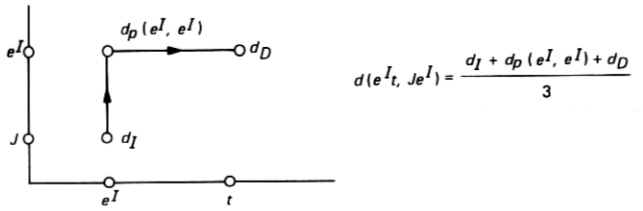
$$d(e^I t, Je^I) = \frac{d_I + d_p(e^I, e^I) + d_D}{3}, \tag{6a}$$

whereas for one and nine, the optimum path is a straight line giving

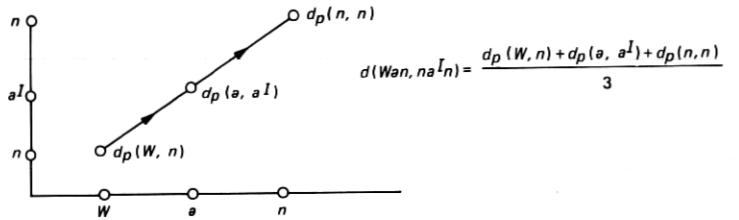
$$d(w\,\text{ə}\,n, na^I n) = \frac{d_p(w, n) + d_p(\text{ə}, a^I) + d_p(n, n)}{3}. \tag{6b}$$

It should be clear that once $d_p(p_1, p_2)$, d_I, and d_D are defined, the word-by-word distance scores can be generated.

A second approach to obtaining word-by-word distance scores is to use real tokens of the vocabulary words and do the actual dynamic time warping of the feature sets and obtain actual word distances. If several tokens have been recorded, averaging of distances increases the reliability of the final results.



(a)



(b)

Fig. 4—Examples illustrating "word" alignment based on dynamic time warping.

From the word-by-word distance matrices, word equivalence classes may be obtained using the clustering procedures of Levinson et al.,[14] in which the vocabulary words are grouped into clusters (equivalence sets) based entirely on pairwise distance scores.

As an example of the use of the above techniques, consider the 39-word vocabulary consisting of the 26 letters of the alphabet, the 10 digits, and the 3 command words STOP, ERROR, and REPEAT. These 39 words become clustered into the sets

		Tokens
ϕ_1	= {B, C, D, E, G, P, T, V, Z, 3, REPEAT},	11
ϕ_2	= {A, J, K, 8, H},	5
ϕ_3	= {F, S, X, 6},	4
ϕ_4	= {I, Y, 5, 4},	4
ϕ_5	= {Q, U, 2},	3
ϕ_6	= {L, M, N},	3
ϕ_7	= {O},	1
ϕ_8	= {R},	1
ϕ_9	= {W},	1
ϕ_{10}	= {STOP},	1
ϕ_{11}	= {ERROR},	1
ϕ_{12}	= {0},	1
ϕ_{13}	= {1},	1
ϕ_{14}	= {7},	1
ϕ_{15}	= {9}.	1

We discuss this vocabulary and the resulting equivalence sets a great deal more in Section III.

2.2 Determination of class-distance scores

Once all the vocabulary words have been assigned to one of the J classes, the first recognition pass estimates an ordering of the word classes in terms of class-distance scores. The class-distance scores can be determined in one of two ways. First they can be computed as the minimum of the word-distance scores, for all words in the class, i.e.,

$$\tilde{d}(\phi_j) = \min_{v_i \in \phi_j} D(\tilde{\mathbf{T}}, \tilde{\mathbf{R}}_i), \qquad j = 1, 2, \cdots, J. \tag{7}$$

This computation is similar to the one used by Aldefeld et al.[15] for directory listing retrieval.

An alternative method of obtaining class-distance scores would be to obtain "class-reference" templates (as well as word-reference templates) and to measure distance directly from the class-reference templates. Clearly with multiple templates per class, the K-nearest neighbor (KNN) rule can be used as effectively for class templates as for word templates.

The reason for considering class-reference templates for obtaining the class-distance scores is that the number of word classes is clearly smaller than the number of words. Hence, the number of distance calculations required to establish class-distance scores is generally much lower for class templates than for word templates. For example, for the 39-word vocabulary discussed previously, there are 15 word classes. Hence there is almost a 3 to 1 reduction from words to word classes. However, it should be clear that the danger in using class templates is that errors in determining class distances can be made from the reduced number of templates. This point will be discussed later in this paper.

2.3 Choice of weighting functions for the second pass of recognition

The output of the first recognition pass is an ordered set of word class-distance scores. For the second recognition pass, all words *within* the top class (or classes) are compared to the unknown test-word pattern ($\tilde{\mathbf{T}}$) using a weighted distance of the type discussed in eq. (5), and an ordering of words *within* the class is made. If several classes have similar class distance scores, the words within each of these classes are ordered in the same manner.

The key question that remains is how do we choose the weighting function, $W(k)$, of eq. (5) in an optimal or reasonable manner. The reader should recall, at this point, that the optimal weighting function, $W(k)$, is assumed to be a function of the pair of indices i (the reference

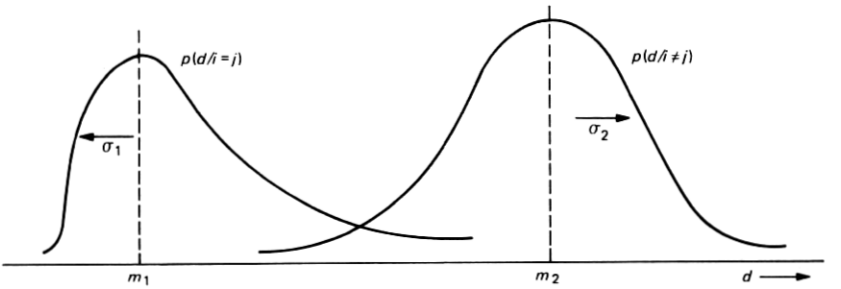


Fig. 5—Simple Gaussian model for frame distance distributions.

word) and j (the proposed test word). Hence if there are L words in an equivalence class, then there are $L(L-1)$ sets of weighting functions [the cases $i = j$ have $W(k) = 1$].

We have investigated two ways of determining $W(k)$ for the second recognition pass. Optimality theory says that to maximize the weighted distance of eq. (5),[16] the value of $W(k)$ should be

$$W(k) = 1 \qquad k = k_0, \tag{8a}$$

$$= 0 \qquad \text{all other } k, \tag{8b}$$

where k_0 is the index where the distance between $\tilde{\mathbf{R}}_i$ and $\tilde{\mathbf{T}}$ is, on average, the maximum. In this manner, the algorithm places all its reliance on the single frame where one would expect the maximum difference between reference and test patterns to occur. In practice, this weighting does not work since the variability in location of the frame $k = k_0$ of eq. (7) is large. Hence, on several trials the distances, using the weighting of eq. (7), can vary considerably.

A more effective manner of determining a good (but not optimal) set of weights is as follows. Consider the model for the distribution of distances for a single frame as shown in Fig. 5. The curve on the left in Fig. 5 is the assumed distribution of distances in the case when $i = j$ (i.e., the reference and test patterns are from the same word). In this case, we expect a χ^2 distribution with p (order of the LPC model) degrees of freedom for the frame distance. For convenience, we model this distribution as a Gaussian distribution with mean m_1 and standard deviation σ_1.*

For the case when $i \neq j$ (i.e., the reference and test patterns are from different words), we assume the frame distance has a Gaussian distri-

* This assumption is reasonable since the word distance, which is a sum of frame distances, has a Gaussian distribution (by the central limit theorem), and the actual probability of word error is directly related to the word distance.

bution (as shown to the right in Fig. 5) with mean m_2 and standard deviation σ_2.

We now make a simple recognition model that says the probability of recognition error for the word is proportional to the probability of error for single frames (since the word distance is the sum of frame distances). Then, based on the model of Fig. 5 with assumed Gaussian statistics, the probability of correct classification (i.e., finding a smaller frame distance for the spoken word, than for any other word) for a single frame is

$$P(C) = \int_{-\infty}^{\infty} P[p(d/_{i=j}) = \lambda] \cdot P[p(d/_{i\neq j}) > \lambda]\, d\lambda, \tag{9}$$

where $P[x]$ is the probability of the event x occurring. Equation (9) says that the probability of correct frame classification is the integral of the probability that for the correct word ($i = j$) we get a frame distance λ, and for the *closest* incorrect word ($i \neq j$) we get a frame distance greater than λ. Thus the probability of a frame error is

$$P(E) = 1 - P(C), \tag{10}$$

which becomes

$$P(E) = 1 - \int_{-\infty}^{\infty} N[\lambda - m_1, \sigma_1] \int_{\lambda}^{\infty} N[\eta - m_2, \sigma_2]\, d\eta\, d\lambda, \tag{11}$$

which can be put into the form

$$P(E) = \int_{-\infty}^{(m_2-m_1)/(\sigma_1^2+\sigma_2^2)^{1/2}} \frac{\exp(-x^2/2)}{\sqrt{2\pi}} dx = Erf\left(\frac{m_2 - m_1}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right). \tag{12}$$

The form of eq. (12) can be verified for the simple cases $m_2 = m_1$, where $P(E) = 0.5$, and $m_2 \gg m_1$, where $P(E) \to 0$.

The above discussion suggests that a reasonable choice for frame weighting would be

$$W^{j,i}(k) = \frac{|\langle \hat{d}_{ii}(k)\rangle - \langle \hat{d}_{ji}(k)\rangle|}{(\sigma^2_{\hat{d}_{ii}(k)} + \sigma^2_{\hat{d}_{ji}(k)})^{1/2}}, \tag{13}$$

where $\hat{d}_{ii}(k)$ is the local distance between repetitions of word i for frame k, and $\hat{d}_{ji}(k)$ is the local distance between spoken words j and i for frame k, and where the expectations are performed statistically over a large number of occurrences of the words v_i and v_j.

By way of example, Fig. 6 shows examples of plots of $\langle d_{ji}(k)\rangle$ versus

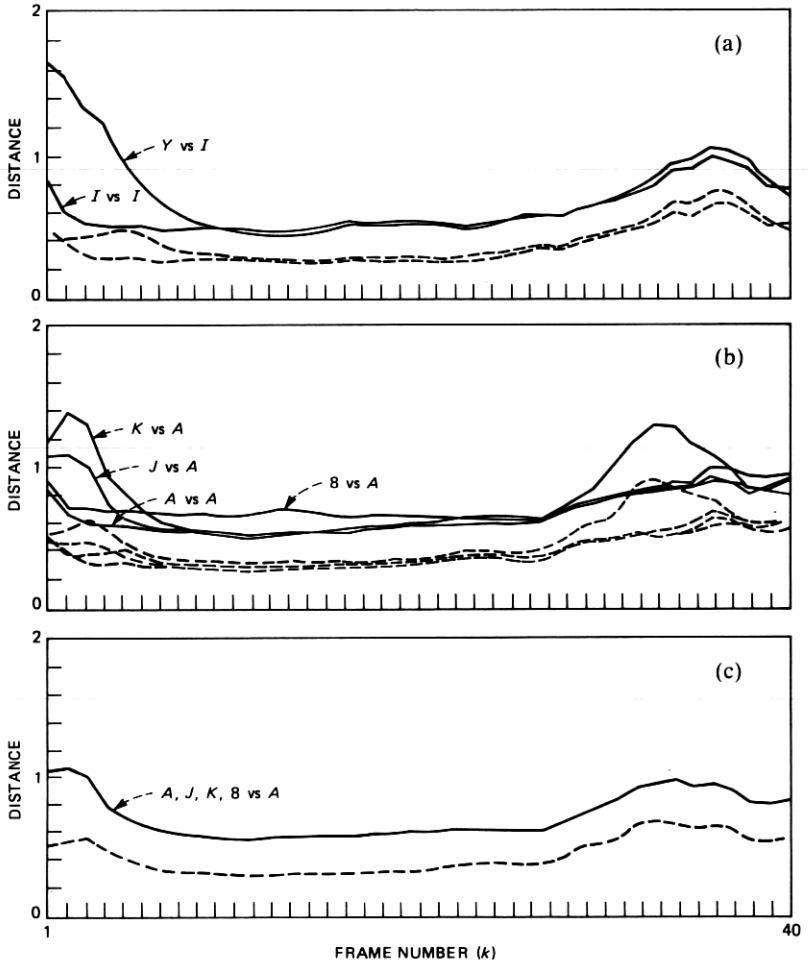


Fig. 6—Examples of frame-by-frame distances for words within word equivalence classes.

k and $W(k)$ versus k for some typical cases.* Figure 6 shows a series of plots for the following cases:

(i) (Fig. 6a) Curves of $\langle \hat{d}_{ji}(k) \rangle$ and $\sigma_{\hat{d}_{ji}(k)}$ for the case where word i was the letter I, and word j was the letter Y. We can see that $\langle \hat{d}_{ii}(k) \rangle$ (the solid curves) is approximately constant whereas $\langle \hat{d}_{ji}(k) \rangle$ differs from $\langle \hat{d}_{ii}(k) \rangle$ only at the beginning of the word (i.e., the first eight

* The data of Fig. 6 were obtained from about 10,000 comparisons for each word, i.e., a large data base was used.

frames). We also see that the curves of $\sigma_{\hat{d}_{ji}(k)}$ (the dashed curves) are comparable for the cases $j = i$ and for $j \neq i$, with only small differences occurring in the first eight frames.

(*ii*) (Fig. 6b) Curves of $\langle \hat{d}_{ji}(k) \rangle$ and $\sigma^2_{\hat{d}_{ji}(k)}$ for the case where word i was the letter A, and where j corresponded to the letters J and K for word 8. Similar behavior to that of Fig. 6a is seen, in that $\langle \hat{d}_{ii}(k) \rangle$ is approximately constant, and $\langle \hat{d}_{ji}(k) \rangle$ is larger than $\langle \hat{d}_{ii}(k) \rangle$ at the beginning of the word, for words J and K, and at the end of the word, for word 8. For the word 8, the curve of $\sigma_{\hat{d}_{ji}(k)}$ is also fairly large at the end of the word, indicating the high degree of variability in the plosive release of the word 8.

(*iii*) (Fig. 6c) The part shows the results of averaging the data of Fig. 6b over all $j \neq i$ with j in the class of word i, i.e., class-weighting templates. In this case the curve of $\langle \hat{d}_{ji}(k) \rangle$ shows flat behavior except at the beginning (due to J, K) and end (due to 8). If storage of word-weighting curves is burdensome, the use of class-weighting curves could be considered as a viable alternative.

Figure 7 shows a set of two weighting curves $W^{j,i}(k)$ for the words I and Y. Figure 7a shows the weighting curve for reference word I and test word Y, and Fig. 7b shows the weighting curve for reference word Y and test word I. Several interesting properties of the curves should be noted. First we see that $W^{j,i}(k)$ generally consists of a large pulse (for these examples this occurs near $k = 1$) and a residual tail. The tail is a measure of the statistical noise level, i.e., the statistical difference between $\langle \hat{d}_{ji}(k) \rangle$ and $\langle \hat{d}_{ii}(k) \rangle$ in the region of acoustical similarity. Typically the peak amplitude in the tails is less than 10 percent of the peak amplitude in the main pulse.

Another interesting property of the weighting curves is that there is no symmetry, in that

$$W^{i,j}(k) \neq W^{j,i}(k). \tag{14}$$

An explanation of this behavior is given in Fig. 8, which shows two plots of dynamic time-warping paths for the words I and Y, where it is assumed that the word Y is simply the word I with a prefix phoneme $/w/$. Figure 8a shows that when I is warped to Y, there is a discrepancy region in which the $/w/$ is being warped to the initial region of the $/a^I/$ and large distances result. The $/a^I/$ is warped to itself (the "ideal" path) and no further distance is accumulated. Figure 8b shows that the discrepancy region is considerably smaller when Y is mapped to I. The resulting weighting curves agree in form with the results given in Fig. 7.

2.4 Generation of distance scores for the second recognition pass

We have now shown how to assign words to classes, how to get class

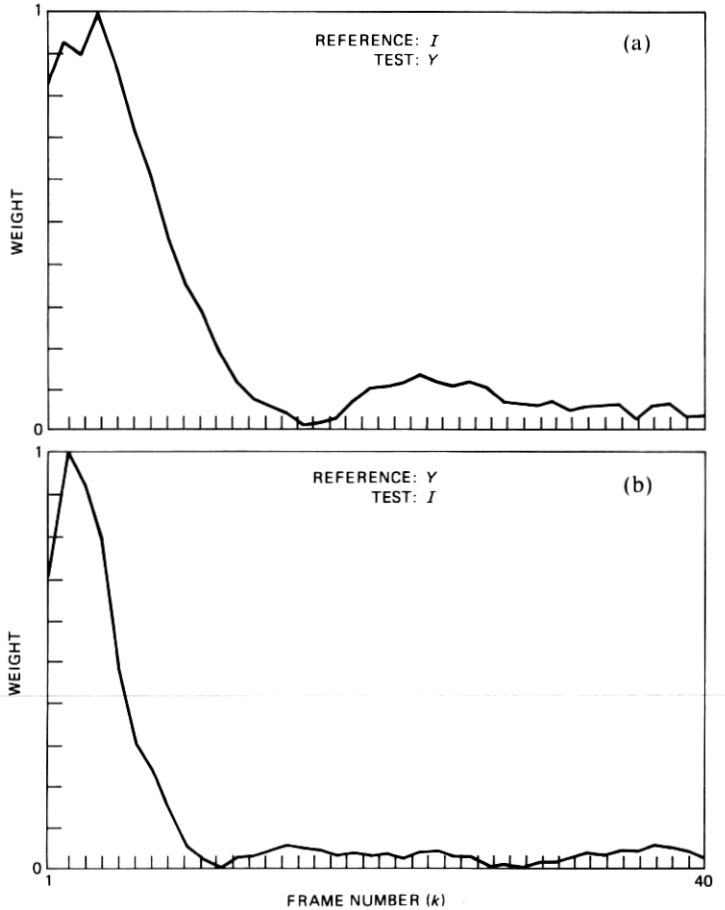


Fig. 7—Weighting curves for comparing the words / *I* / and / *Y* /.

distance scores for the first recognition pass, and how to assign weights for pairs of words within a word class. The next step in the procedure is the determination of the distance for the second recognition pass based on the pairwise weighted distance scores.

To see how this is accomplished, we define a pairwise weighted distance $D_{j,i}$ as

$$D_{j,i} = \frac{\sum_{k=1}^{N} W^{j,i}(k)\, d_i(k)}{\sum_{k=1}^{N} W^{j,i}(k)}, \tag{15}$$

where i is the index of the reference pattern (i.e., one of the words in

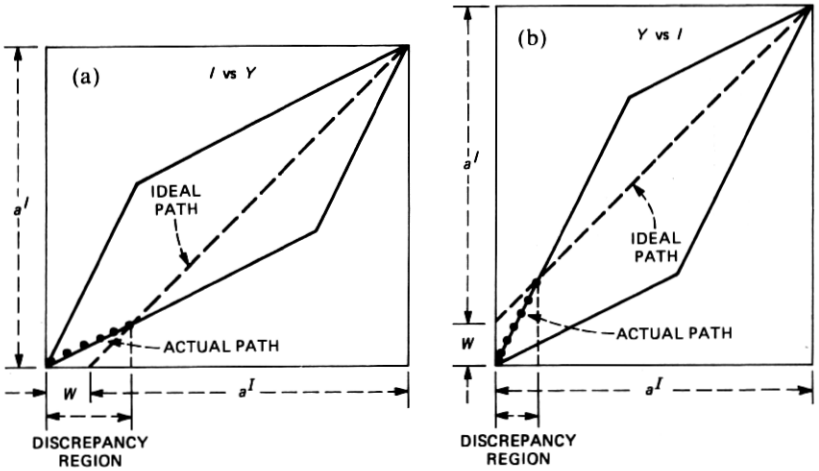


Fig. 8—An example showing why the word weighting curves are not symmetrical.

the equivalence class) and j is the (assumed) index of the test pattern (again one of the words in the equivalence class).

The quantity $D_{j,i}$ of eq. (15) is computed for all i, j pairs (with $i \neq j$) in the word class with minimum class distance, and a matrix of pairwise distances $\mathbf{D}$ is obtained. The word distance, D_i, can be obtained in one of two ways, namely:

(i) Averaging over the j index, giving

$$D_i = \sum_{\substack{j \\ j \neq i}} D_{j,i}. \tag{16a}$$

(ii) Finding the minimum over the j index, i.e.,

$$D_i = \min_{\substack{j \\ j \neq i}} \{D_{j,i}\}. \tag{16b}$$

The advantage of averaging is that D_i tends to be more reliable, since averaging is equivalent to adding weighted distances over a larger number of frames than would be used for a single comparison. The minimum computation is useful, especially when several of the $D_{j,i}$ are about the same. We examine both these scoring methods in Section III.

For the case of averaging pairwise distance scores [eq. (16a)], the computation can be carried out more efficiently as follows. By combining eqs. (15) and (16a) we get

$$D_i = \sum_j D_{j,i} = \sum_j \left(\frac{\sum_{k=1}^{N} W^{j,i}(k)\, d_i(k)}{\sum_{k=1}^{N} W^{j,i}(k)} \right) \tag{17a}$$

$$= \sum_j \sum_{k=1}^{N} \left(\frac{W^{j,i}(k)\, d_i(k)}{\sum_{k=1}^{N} W^{j,i}(k)} \right) \tag{17b}$$

$$= \sum_{k=1}^{N} \sum_j \left(\frac{W^{j,i}(k)}{\sum_{k=1}^{N} W^{j,i}(k)} \right) d_i(k) \tag{17c}$$

$$= \sum_{k=1}^{N} \hat{W}^i(k)\, d_i(k), \tag{17d}$$

where

$$\hat{W}^i(k) = \sum_j \frac{W^{j,i}(k)}{\sum_{k=1}^{N} W^{j,i}(k)}. \tag{18}$$

Thus, for L words in the equivalence class, we can compute D_i with N multiplications and additions [rather than the $N(L-1)$ computations of eq. (16a)], and only L vectors of N averaged weights $[\hat{W}^i(k)]$ need be stored, rather than $L(L-1)$ vectors as implied by eq. (15).

Another variation on the distance weighting that was studied here was the effect of applying a nonlinearity to the weighting function, $W^{j,i}$, before computing $D_{j,i}$. The nonlinearity was to replace $W^{j,i}(k)$ by $\tilde{W}^{j,i}(k)$, defined as

$$\tilde{W}^{j,i}(k) = \begin{cases} W^{j,i}(k) & \text{if } W^{j,i}(k)/W_{\text{MAX}} > T, \\ 0 & \text{otherwise,} \end{cases} \tag{19}$$

where

$$W_{\text{MAX}} = \max_k [W^{j,i}(k)], \tag{20}$$

and T is a threshold which is specified in the algorithm. The nonlinearity of eq. (19) truncates (to 0) the weighting curve whenever its relative amplitude falls below the threshold. Figure 9 illustrates a typical curve $W^{j,i}(k)$ and its truncated version $\tilde{W}^{j,i}(k)$. The new weighting function was then applied directly in eq. (15) in place of $W^{j,i}(k)$. Clearly, when $T = 0$, $W^{j,i}(k)$ and $\tilde{W}^{j,i}(k)$ are identical. Again, when averaging is used, the computation of eq. (17) gives a reduced set of weights.

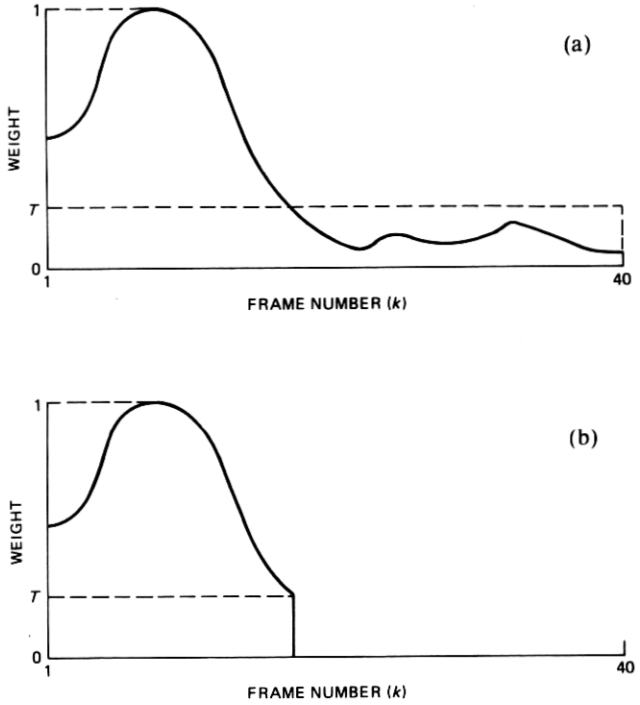


Fig. 9—An example of a weighting curve and its truncated version.

2.5 *Overall distance computation*

If we can make the assumption that the probability of a class error on the first recognition pass is significantly smaller than the probability of a word error on the first pass, then the final distance for each word of the minimum class is the distance obtained on the second recognition pass. However there are applications in which it is desirable to have a distance score for *every* word in the vocabulary. Hence, in these cases, it is necessary to combine the ordering from the second pass, with the distances from the first pass. The basis for such a strategy is that distances on the first pass are statistically more reliable than distances on the second pass, whereas order statistics (within the class) are more reliable on the second pass than on the first pass. One very simple way of combining distances and word orders is to obtain second-pass ordering for every word in the vocabulary (i.e., apply the method of Section 2.4 to all word classes), and then reorder the word list using distances from the first pass, and ordering within the class from the second pass.

2.6 *An example of the use of the two-pass system*

To illustrate this entire procedure, Tables I to III show an example

Table I—Recognition results for a simple example (first pass)

Word Index	Word Class	Word Distance First Pass	Word Position First Pass	Class Number	Class Distance First Pass
1	1	0.47	4	1	0.47
2	3	0.39	2	2	0.66
3	3	0.51	5	3	0.37
4	2	0.72	10		
5	3	0.42	3		
6	1	0.60	6		
7	1	0.67	9		
8	2	0.83	12		
9	3	0.37	1		
10	2	0.78	11		
11	2	0.66	8		
12	1	0.62	7		

of the recognition steps for a 12-word vocabulary with three word equivalence classes. Table I shows the results of the first recognition pass. The class distance scores are assigned as the minimum word distance for words within the class. The "best" class in the first pass is class 3 with a distance score of 0.37, with class 2 having a somewhat higher distance of 0.47. In the second recognition pass the words within the best class (or classes) are compared using the optimally determined

Table II—Second recognition pass results for the example in Table I

		j = 1	j = 6	j = 7	j = 12	D_i(avg)	Order	
i	1	X	0.43	0.52	0.47	0.47	1	
	6	0.57	X	0.62	0.62	0.60	3	Class 1
	7	0.72	0.75	X	0.60	0.69	4	
	12	0.60	0.57	0.63	X	0.60	2	
		$D_{j,i}$						

		j = 4	j = 8	j = 10	j = 11	D_i(avg)	Order	
i	4	X	0.87	0.82	0.85	0.85	3	
	8	0.80	X	0.84	0.86	0.83	2	Class 2
	10	0.92	0.77	X	0.91	0.87	4	
	11	0.78	0.80	0.80	X	0.79	1	
		$D_{j,i}$						

		j = 2	j = 3	j = 5	j = 9	D_i(avg)	Order	
i	2	X	0.33	0.25	0.28	0.29	1	
	3	0.47	X	0.67	0.50	0.55	4	Class 3
	5	0.45	0.56	X	0.57	0.53	3	
	9	0.27	0.37	0.30	X	0.31	2	
		$D_{j,i}$						

weighting functions. The results for each of the three classes are shown in Table II. In practice, one would usually need to compute the $D_{j,i}$ scores only for the best one or two classes. However, for explanatory purposes, results are shown for all three classes. Also, as discussed above, in the case of distance averaging, the $D_{j,i}$ scores need not be computed since the D_i scores can be obtained directly via eqs. (17) and (18). Using the technique of averaging leads to the within-class distances and orderings as shown in the table. Finally, Table III shows the results of reordering the words using the distances obtained from pass 1, and the within-class orderings obtained from pass 2. Thus word 2 is the best recognition candidate (with a distance of 0.37), whereas word 9 was the best recognition candidate at the end of the first pass. Other, within-class reshufflings of word position occur as a result of the two recognition passes as shown in Table I.

2.7 *Summary of the two-pass recognizer*

Figure 10 shows a block diagram of the full two-pass isolated word recognition system. In the first pass a DTW distance is computed between the unknown test word and the reference templates for each word class. The outputs of the first pass are ordered sets of word distance scores and class distance scores.

For the second pass a set of pairwise weighted distances is determined for all words within each word class with suitably low scores on the first recognition pass. The final recognition output is a combination of distance scores from the first pass and word orderings from the second pass. In the next section we demonstrate how this procedure works in some practical recognition examples.

Table III—Overall word positions and distances for the example given in Tables I and II

Word Index	Word Position	Word Distance
1	4	0.47
2	1	0.37
3	5	0.51
4	11	0.78
5	3	0.42
6	7	0.62
7	9	0.67
8	10	0.72
9	2	0.39
10	12	0.83
11	8	0.66
12	6	0.60

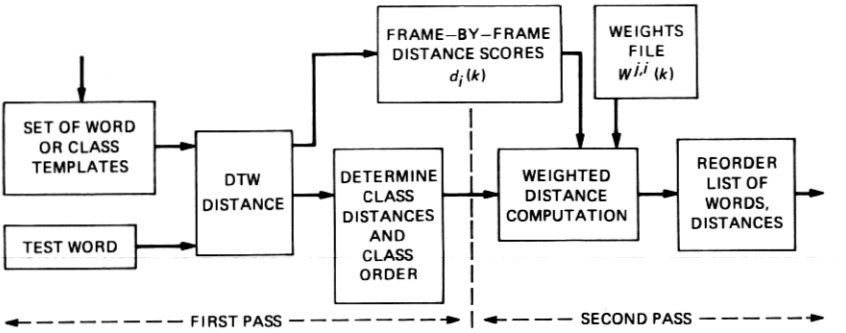


Fig. 10—Block diagram of the overall two-pass recognizer.

III. EVALUATION OF THE TWO-PASS RECOGNIZER

To test the ideas behind the two-pass recognizer, we used a data base of existing recordings. The word vocabulary consisted of the $V = 39$ word vocabulary of the letters of the alphabet, the digits (0 to 9), and the three command words STOP, ERROR, and REPEAT. The training data for obtaining word and class reference templates, and pairwise word weighting curves, consisted of one replication of each word by each of 100 talkers (50 men, 50 women).* The word reference templates (12 per word) were obtained from a clustering analysis of the training data.[14,6] A set of "class" reference templates (12 per class) was obtained from a second clustering analysis in which the words within a class were combined prior to the clustering. The pairwise word weighting curves were obtained by cross-comparing all word tokens within a word class, averaging the time-aligned distance curves, and computing both the averages and standard deviations for each frame.

To test the performance of the overall system, two test sets of data were used. These included:

1. TS1—10 talkers (not used in the training) spoke the vocabulary one time over a dialed-up telephone line.
2. TS2—10 talkers (included in the training) spoke the vocabulary one time over a dialed-up telephone line.

Two sets of performance statistics were measured. For the first recognition pass the ability of the recognizer to determine the correct word class was measured. For the second recognition pass the improvement in word recognition accuracy (over the standard one-pass approach) was measured. The results obtained are presented in the next two sections.

* All results presented here are for speaker independent systems.

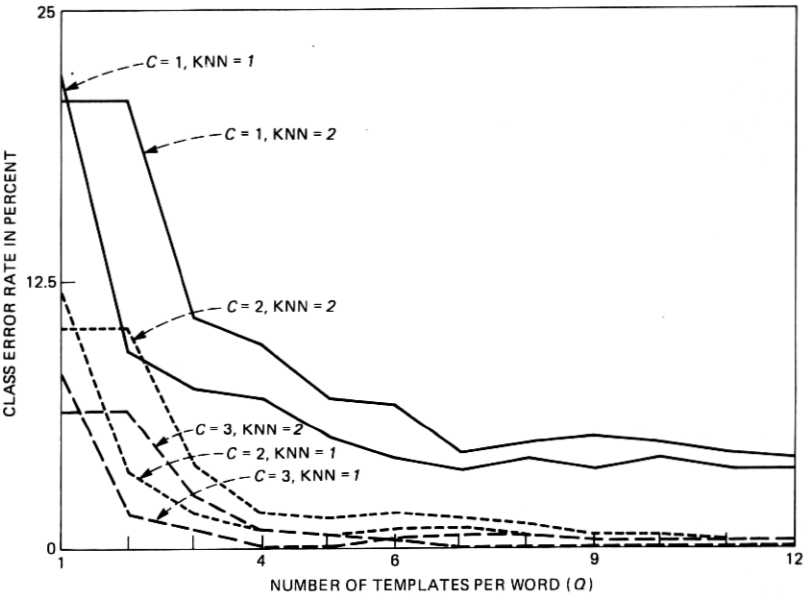


Fig. 11—Plots of class accuracy as a function of the number of templates per word (Q), class position (C), and KNN rule (KNN) for a 15-class vocabulary.

3.1 Class recognition accuracy for the first pass

The ability of the recognizer to determine the "correct" word class of the spoken word was measured using both word templates (and obtaining class-distance scores from the word-distance scores as discussed previously), and class templates (obtaining class-distance scores directly). The number of templates per word (or per class) varied from 1 to 12 in the tests to see the effects of the number of reference templates on the class accuracy. The K-nearest neighbor (KNN) rule was used to measure class scores with values of KNN = 1 (minimum distance), KNN = 2 (average of two best scores), and KNN = Q (average of Q best scores), where Q was the total number of templates used per word (or per class).

The results of the class recognition accuracy tests are given in Figs. 11 and 12.* Figures 11 and 12 show plots of class error rate (based on the top C classes) as a function of the number of templates per word (Fig. 11) or templates per class (Fig. 12), for values of KNN = 1 and 2, and for $C = 1$ (top candidate), $C = 2$ (two best classes), and $C = 3$ (three best classes). Figure 11 shows results when each class is represented by word templates, and Fig. 12 shows results when each class is represented by class templates.

* The reader should note the difference in vertical scales between Figs. 11 and 12.

Several interesting observations can be made from Figs. 11 and 12. These include:

(*i*) The KNN = 1 rule performs consistently better than the KNN = 2 rule for class discrimination, for *all values* of C and Q. This result is in contradiction with the results of Rabiner et al.[6] who found significantly better performance for KNN = 2 than for KNN = 1. The explanation of this behavior is that the KNN = 2 rule provides significantly improved, within-class discrimination (at the expense of slightly worse between class discrimination), and that when the only function is to determine the class, the KNN = 1 rule is superior. In fact when the KNN rule was used with a value of KNN = Q (i.e., averaging over all Q reference templates), the class accuracy on the first candidate decreased by about 20 percent—a highly significant loss of accuracy. This result again demonstrates that the minimum distance rule (KNN = 1) is best for *class* discrimination.

(*ii*) The use of word-reference templates provides significantly better performance than obtained from class-reference templates. For example, the class error rate for the top three classes ($C = 3$) with $Q = 4$ templates per word is essentially 0; whereas the class error rate for the top three classes with four templates per class is about 4 percent. This result shows clearly the importance of representing each word in

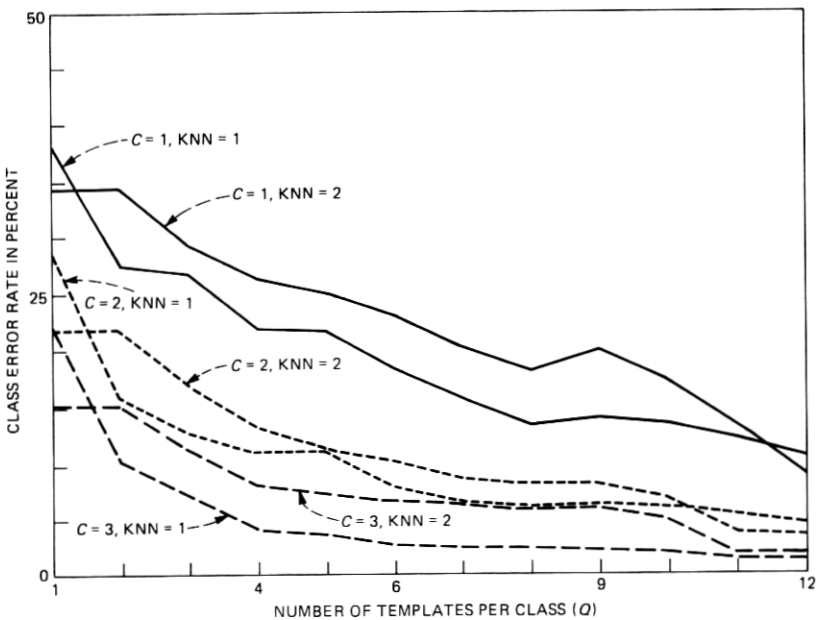


Fig. 12—Plots of class accuracy as a function of the number of templates per class (Q), class position (C), and KNN rule (KNN) for a 15-class vocabulary.

the equivalence class by an adequate number of word-reference templates.

(*iii*) With six templates per word, error rates of about 4 percent ($C = 1$), 1 percent ($C = 2$), and 0 percent ($C = 3$) are obtainable, indicating that the full contingent of 12 templates per word is unnecessary for proper class determination. Using 6, rather than 12 templates per word reduces the computation in the first recognition pass by 50 percent. If we *always* use two or more word classes, the required number of templates per word for the first pass can be reduced to four, with no serious loss in class accuracy.

The results shown in Fig. 11 indicate that high accuracy can readily be achieved in determining the correct equivalence class for each word in a very complex vocabulary. Hence there would appear to be no problems in implementing the first pass of the recognition system.

3.2 Within-class word discrimination for the second pass and overall performance scores

The two-pass word recognizer was tested on the words of TS1 and TS2. For each test set a total of 390 words were used (39 words × 10 talkers). For TS1, the word recognition accuracy (for the best candidate) on the first pass was 78 percent, and for TS2 (with talkers from the training set) the word recognition accuracy on the first pass was 85 percent. At the output of the second pass, the word recognition accuracy for the best candidate [using the averaging technique of eq. (16a) and assuming the correct word equivalence class was found] was 84.6 percent for TS1 and 88.5 percent for TS2, representing potential improvements of 6.6 percent and 3.5 percent, respectively. The reason that a larger improvement in accuracy was obtained for TS1 data than for TS2 data was that the accuracy on the first pass was lower for TS1 than for TS2 (where the talkers were in the training set) and hence there was more room for improvement within the word classes.

Figures 13 and 14 show plots of the changes in accuracy that are obtained for TS1 (Fig. 13) and TS2 (Fig. 14) data when a threshold is imposed on the distance scores at the output of the first recognition pass. The threshold specifies that the second recognition pass is skipped if the distance of the second word candidate is more than the threshold greater than the distance of the first word candidate. Clearly this procedure is a strictly computational one, since low-distance scores for a single word on the first pass are highly reliable indicators that no second pass is necessary. The data plotted in Figs. 13 and 14 show the percentage of cases where the actual spoken word comes in a lower position on the second pass than in the first pass within the word class; it also shows the percentage of cases when the spoken word comes in a higher position on the second pass than the first pass, and the

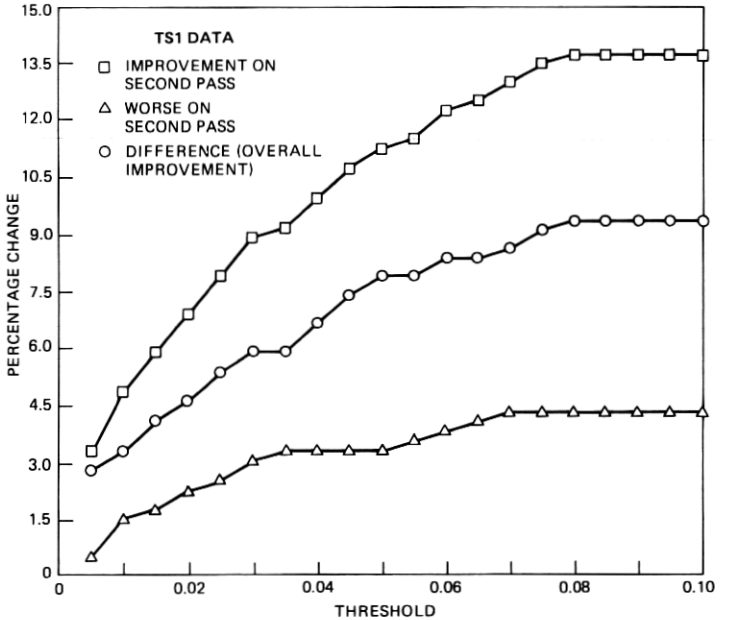


Fig. 13—Percentage improvement, decrease, and the resulting difference in word position at the output of the second recognition pass for TS1 data as a function of the distance threshold using the averaging method.

difference (the improvement) between the two curves. All the results are plotted as a function of the distance threshold for performing the second-pass computation. It can be seen from these figures that the two-pass recognizer is not ideal, i.e., there is a significant fraction of words for which a worse position results at the output of the second pass. However, on balance, it is seen that a real improvement in recognition accuracy results, and it is this improvement that makes the procedure a viable one.

A similar set of results obtained using the minimum computation of eq. (16b) on the second pass rather than the average computation of eq. (16a) are shown in Figs. 15 and 16 for TS1 and TS2, respectively. These plots show the same information as those of Figs. 13 and 14 for the averaging procedure. A comparison of these results shows that the averaging computation performs as well as, or better than, the minimum computation for the whole range of distance thresholds, and for both data sets. These results indicate that the averaging method provides a small but important statistical stability to the computation.

3.3 The effect of thresholding on the weighting curves

We ran a series of tests with the data from TS1 and TS2 to investigate the effects of applying thresholds to the weighting curves as illustrated

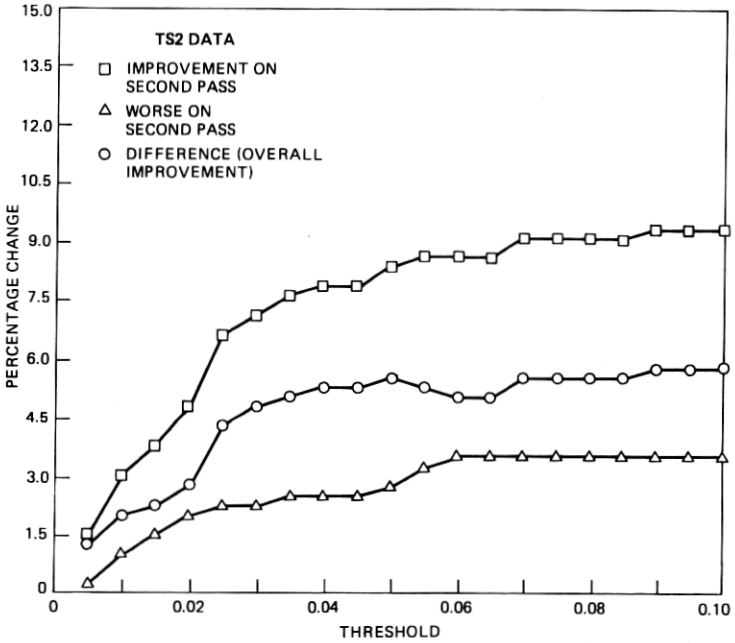


Fig. 14—Percentage improvement, decrease, and the resulting difference in word position at the output of the second recognition pass for TS2 data as a function of the distance threshold using the averaging method.

in Fig. 9. The results indicated that poorer performance *always* resulted when any significant part of the weighting curve was zeroed out. Thus the gain achieved by removing the "statistical" low-level parts of the weighting curve was canceled by the "deterministic" loss from the rest of the weighting curve. Hence the conclusion was to use the entire weighting curve as derived from the statistical model.

3.4 Computation for the two-pass recognizer

We have seen in Section 3.3 that word recognition accuracy improvements of from 3.5 to 6.6 percent result for the 39-word vocabulary using the two-pass recognizer. A key question that must be answered is what is the cost of the computation for the two-pass system.

To answer this question we must examine the computation in each pass of the recognizer. In the first recognition pass, for a V-word vocabulary with Q templates per word, a total of QV DTW comparisons are made. For a value of $N = 40$, each DTW comparison requires about 500 nine-point dot-product computations, so a total rate, R_1, of

$$R_1 = Q \cdot V \cdot 500 \cdot 9 \tag{21}$$

multiplications and additions are required.

If we assume that the local distances $d_{j,i}(k)$ associated with the optimum warping paths are saved for *each* reference template, then for *each* pairwise comparison of the second pass a total of N (typically 40) multiplications and additions are required. For L words in the equivalence class, a total of

$$R_2 = L \cdot (L - 1) \cdot N \tag{22}$$

multiplications and additions are required for the second-pass computation for a single equivalence class. For the averaging procedure of eq. (17), R_2 is reduced to LN multiplications and additions.

If we assume typical values of $V = 39$, $Q = 12$, $L = 7$, $N = 40$, we get $R_1 = 2{,}106{,}000$ and $R_2 = 1680$, i.e., the computation of the second pass is insignificant compared to the first pass computation. Furthermore since we can use reduced values of Q for the first pass (i.e., $Q = 6$ or $Q = 4$) the overall computation can be significantly reduced from the standard isolated word recognizer, with the same improvement in accuracy!

IV. DISCUSSION

The results presented in the preceding section show that improved recognition accuracy can be obtained via a two-pass recognition algorithm. It was shown that the improvements were both global, i.e., in

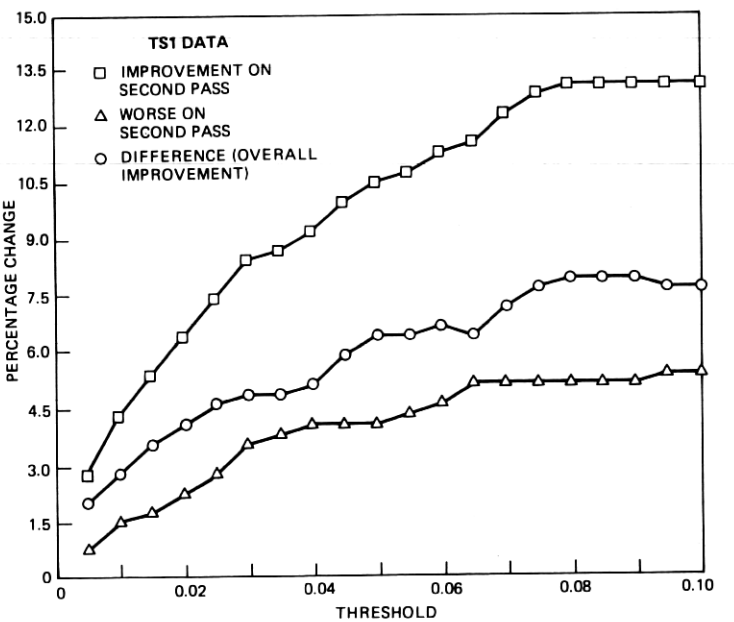


Fig. 15—The same results as in Fig. 13 obtained using the minimum method.

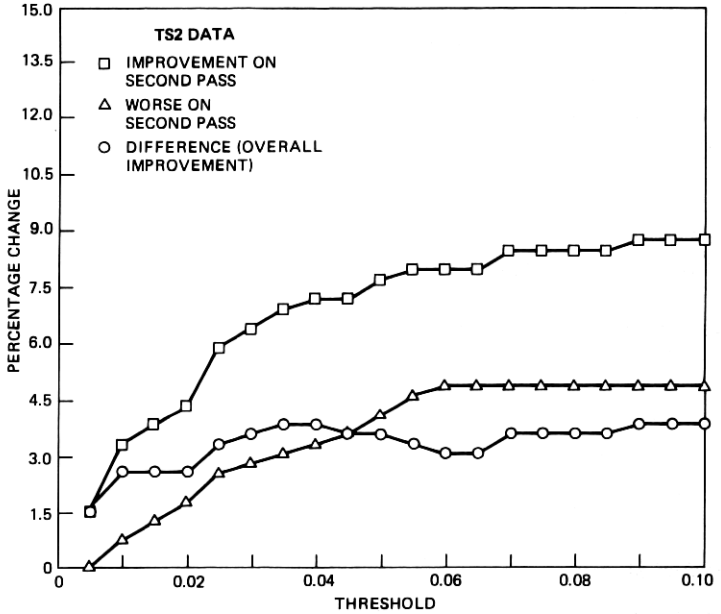


Fig. 16—The same results as in Fig. 14 obtained using the minimum method.

an absolute recognition sense, and local, i.e., within the classes of equivalent words. Although the proposed two-pass recognizer has a number of possible implementations, it was shown that the best choices were to use a reduced set of word templates on the first pass, and to use all word classes that had reasonably small distance scores on the second pass.

One of the major issues that remains unresolved in the two-pass recognizer is the choice of weighting curve used in the second-pass distance computation. The assumed Gaussian model which led to the variance-weighted difference of means for the weights is, at best, an approximation to the actual situation. Experimentation with modified forms of the weighting curve of eq. (13) led to poorer recognition performance. Thus, because we lacked a viable alternative, the weighting curve of eq. (13) is the only one we investigated for use in the two-pass recognizer.

An interesting question that arises as a result of this study is how could this two-pass recognizer aid in practical recognition tasks. As one would anticipate, the answer to this question is that it depends on the specific recognition task. For example, for the backtracking directory listing retrieval system of Rosenberg and Schmidt,[17] the improvement in recognition accuracy could provide significant reductions in

search time. However, for the search procedure of Aldefeld et al.,[15] the increased word accuracy would have no effect on the search time, but could increase the name accuracy, especially when similar names exist in the directory (e.g., T. Smith and P. Smith). For applications like the airlines reservation system of Levinson and Rosenberg,[18] the increased word accuracy would reduce the load on the syntax analyzer; however, it needn't necessarily increase the overall accuracy of the system.

The above examples show that the two-pass recognition strategy can be useful for some applications, but one must examine carefully the specific task before claiming how useful it will potentially be.

V. SUMMARY

We have shown that a two-pass approach to isolated word recognition is viable when the word vocabulary consists of sets of acoustically similar words. The first recognition pass attempts to determine accurately the class within which the spoken word occurs, and the second recognition pass attempts to order the words within the class, based on weighted distances of pairwise comparisons of all words within the class. We discussed several alternatives for implementing this two-pass recognizer, and we made a performance evaluation which showed that a reliable class decision could be made based on a reduced set of template scores, and an improved word decision could be made from weighted pairwise distance scores.

REFERENCES

1. T. B. Martin, "Practical Applications of Voice Input to Machine," Proc. IEEE, *64* (April 1976), pp. 487–501.
2. S. Moshier, "Talker Independent Speech Recognition in Commercial Environments," Speech Communication Papers at the 97th ASA Meeting, June 1979, pp. 551–3.
3. H. Sakoe, "Two-Level DP Matching—A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Proc., *ASSP-27,* No. 6 (December 1979), pp. 588–95.
4. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoustics, Speech, and Signal Proc., *ASSP-23,* No. 1 (February 1975), pp. 67–72.
5. M. R. Sambur and L. R. Rabiner, "A Speaker-Independent Digit-Recognition System," B.S.T.J., *54,* No. 1 (January 1975), pp. 81–102.
6. L. R. Rabiner, S. E. Levinson, A. E. Rosenberg, and J. G. Wilpon, "Speaker-Independent Recognition of Isolated Words Using Clustering Techniques," IEEE Trans. Acoustics, Speech, and Signal Proc., *ASSP-27,* No. 4 (August 1979), pp. 336–49.
7. J. S. Bridle and M. D. Brown, "Connected Word Recognition Using Whole Word Templates," Proc. Inst. Acoustics, Autumn Conf., 1979.
8. C. S. Myers and L. R. Rabiner, "Connected Digit Recognition Using a Level Building DTW Algorithm," IEEE Trans. Acoustics, Speech, and Signal Proc., *ASSP-29,* No. 3 (June 1981).
9. B. Gold, "Word Recognition Computer Program," MIT, RLE Tech. Report 452, June 1966.
10. C. S. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," IEEE Trans.

Acoustics, Speech, and Signal Proc., *ASSP-28*, No. 6 (December 1980), pp. 622–35.
11. P. V. de Souza, "Statistical Tests and Distance Measures for LPC Coefficients," IEEE Trans. Acoustics, Speech, and Signal Proc., *ASSP-25*, No. 6 (December 1977), pp. 554–9.
12. J. M. Tribolet, L. R. Rabiner, and M. M. Sondhi, "Statistical Properties of an LPC Distance Measure," IEEE Trans. Acoustics, Speech, and Signal Proc., *ASSP-27*, No. 5 (October 1979), pp. 550–8.
13. N. Chomsky and M. Halle, *The Sound Pattern of English*, New York: Harper and Row, 1968.
14. S. E. Levinson, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "Interactive Clustering Techniques for Selecting Speaker-Independent Reference Templates for Isolated Word Recognition," IEEE Trans. Acoustics, Speech, and Signal Proc., *ASSP-27*, No. 2 (April 1979), pp. 134–41.
15. B. Aldefeld, L. R. Rabiner, A. E. Rosenberg, and J. G. Wilpon, "Automated Directory Listing Retrieval System Based on Isolated Word Recognition," Proc. IEEE, *68*, No. 11 (November 1980), pp. 1364–79.
16. G. S. Sebestyen, *Decision Making Processes in Pattern Recognition*, New York: MacMillan, 1962.
17. A. E. Rosenberg and C. E. Schmidt, "Automatic Recognition of Spoken Spelled Names for Obtaining Directory Listings," B.S.T.J., *58*, No. 8 (October 1979), pp. 1797–823.
18. S. E. Levinson and A. E. Rosenberg, "Some Experiments With a Syntax-Directed Speech Recognition System," Proc. Int. Conf. Acoust., Speech, and Signal Proc. (April 1978), pp. 700–3.