THE BELL SYSTEM TECHNICAL JOURNAL

DEVOTED TO THE SCIENTIFIC AND ENGINEERING ASPECTS OF ELECTRICAL COMMUNICATION

Volume 60 March 1981 Number 3

On the Use of Dynamic Time Warping for Word Spotting and Connected Word Recognition*

By C. S. MYERS, L. R. RABINER, and A. E. ROSENBERG

(Manuscript received September 9, 1980)

Several variations on algorithms for dynamic time warping for speech processing applications have been proposed. This paper compares two of these algorithms, the fixed-range method and the local minimum method. We show that, based on results from some simple word spotting and connected word recognition experiments, the local minimum method performs considerably better than the fixed-range method. We describe explanations of this behavior and techniques for optimizing the parameters of the local minimum algorithm for both word spotting and connected word recognition.

I. INTRODUCTION

Time registration of a test and a reference pattern is one of the fundamental problems in the area of automatic speech recognition. This problem is important because the time scales of a test and a reference pattern are not perfectly aligned. In some cases the time scales can be registered by a simple linear compression or expansion[1,2]; however, in most cases, a nonlinear time warping is required to compensate for local compression or expansion of the time scale. For such cases, the class of algorithms known as dynamic time warping (DTW) methods has been developed. Work by Sakoe and Chiba,[3]

* The work presented here is based, in part, on the MS thesis, "A Comparative Study of Several Dynamic Time Warping Algorithms for Speech Recognition," by C. S. Myers, MIT, April 1980.

Itakura,[4] and White and Neely[2] has shown that DTW algorithms are an effective method of time registering patterns in isolated word recognition systems. Bridle[5] and Christiansen and Rushforth[6] have studied the applicability of DTW algorithms to word spotting, and recently, Sakoe,[7] Rabiner and Schmidt,[8] and Myers and Rabiner,[9] have successfully applied dynamic time-warping techniques to connected digit recognition. A great deal of work has been done in the area of performance evaluation of the various DTW algorithms as applied to discrete word recognition.[10–12] However, the effects of the DTW parameters on the overall performance of the algorithm for either word spotting or connected word recognition are not as well understood. The purpose of this paper is to discuss several proposed methods of applying DTW algorithms to word spotting and connected word recognition, and to study some of the factors which determine the performance of these algorithms.

The organization of this paper is as follows. In Section II we review the basic dynamic programming method of time alignment and show how it may be used efficiently in either a word spotting or a connected word recognition problem. We describe, in detail, two different DTW algorithms for which we have performed extensive evaluations. Section III contains a description of the experiments which we performed to evaluate the performance of the different DTW algorithms and the effects of the parameters associated with them. In Section IV we summarize the results of these experiments and draw some general conclusions on the use of DTW algorithms for word spotting and connected word recognition.

II. DYNAMIC PROGRAMMING FOR TIME ALIGNMENT

In this section we first review the basic principles of DTW algorithms as applied to discrete word recognition, and then point out some of the inherent difficulties involved in applying these algorithms to word spotting and connected speech recognition. We then show how it is possible to modify the basic DTW idea so that it may be used for both connected word recognition and word spotting applications.

2.1 Dynamic time warping for discrete word recognition

The problem of time alignment for discrete word recognition is illustrated in Fig. 1. A reference pattern, $\mathbf{R}(n)$, $n = 1, 2, \cdots, N$, consisting of a time sequence (i.e., frames) of a multidimensional feature vector is to be time registered with a test pattern, $\mathbf{T}(m)$, $m = 1, 2, \cdots, M$, which is also represented as a time sequence of a multidimensional feature vector. In Fig. 1, for the sake of clarity, both $\mathbf{R}(n)$ and $\mathbf{T}(m)$ are shown as one-dimensional functions. We shall assume that both the reference and the test pattern are measured from

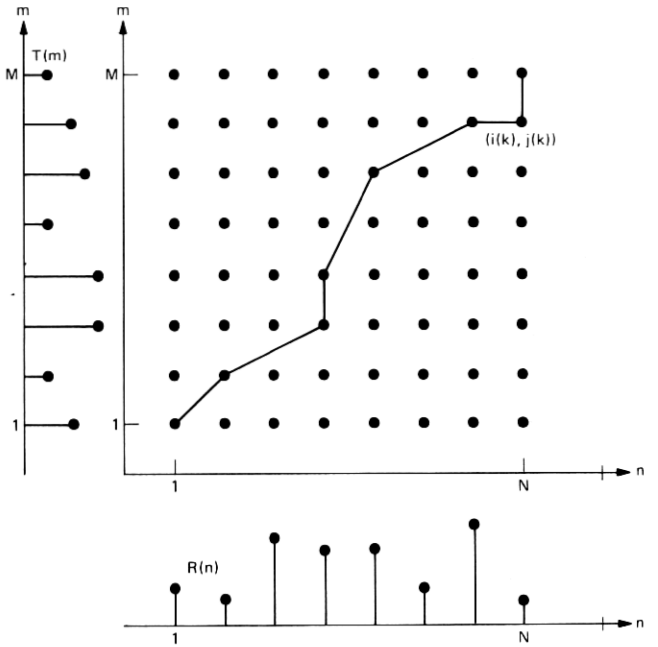

Fig. 1—Time warping of a reference and a test pattern.

the acoustic waveform of a single word, spoken in isolation, and that both the beginning and ending points of the reference and the test pattern have been accurately determined. The problem of time alignment is to find the path, here parameterized by the function pair $(i(k), j(k))$, which minimizes a given distance metric. A typical distance metric* is of the form

$$D(i(k), j(k)) = \frac{\sum_{k=1}^{K} d(i(k), j(k))\bar{W}(k)}{\mathbf{N}(\bar{W})}, \tag{1}$$

where K is the length of the path, $d(i(k), j(k))$ is the local distance, or dissimilarity, between frame $i(k)$ of the reference pattern and frame $j(k)$ of the test pattern, $\bar{W}(k)$ is a weighting function applied to the path, and $\mathbf{N}(\bar{W})$ is a normalization factor which is based on the particular weighting function that is chosen.

In addition to minimizing the global distance, the time alignment path is chosen to have certain desirable properties. One important property is the proper time registration of the beginning and ending points of the test and reference patterns, i.e.,

* D is shown here as a functional of the path function pair $(i(k), j(k))$.

$$i(1) = 1, \qquad j(1) = 1, \tag{2a}$$

$$i(K) = N, \qquad j(K) = M. \tag{2b}$$

Also, the time alignment path is required to obey certain shape and slope constraints. For example, it would not be reasonable to allow a path for which a 10 to 1 expansion or compression of the time axis occurs. Another consideration is the preservation of time order, i.e., the functions $i(k)$ and $j(k)$ must both be monotonically increasing.

These local continuity constraints are generally described by specifying the full path in terms of simple local paths which may be pieced together to form larger paths. For example, to reach a grid print (n, m) it may be reasonable to have come from any of the grid points $(n - 1, m - 1)$, $(n - 1, m - 2)$, or $(n - 2, m - 1)$, as shown in Fig. 2, part a. We refer to these constraints as Type I local constraints. Some other proposed sets of local constraints are shown in parts b, c, and d of Fig. 2. The crossed out arc in part d signifies the restriction that a path may not move horizontally for two consecutive segments.[4] All these local constraints limit the overall slope of the time alignment contour

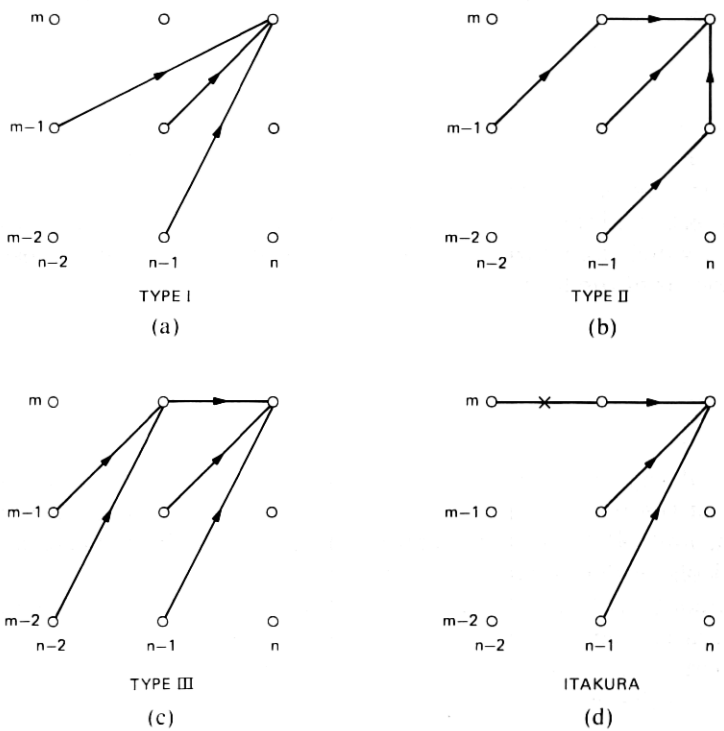

Fig. 2—Local constraints used for dynamic time warping.

to be between ½ and 2, in accordance with the results found by Sakoe and Chiba.[9]

To solve for the optimal time-alignment path, both the weighting function, $\tilde{W}(k)$, and the normalization factor, $\mathbf{N}(\tilde{W})$, must be specified in addition to the local constraints. Typically $\tilde{W}(k)$ is chosen to be either of two functions, i.e.,

$$\tilde{W}(k) = i(k) - i(k-1) \qquad \text{(Type a)}, \tag{3a}$$

$$\tilde{W}(k) = i(k) - i(k-1) + j(k) - j(k-1) \qquad \text{(Type b)}. \tag{3b}$$

These two weighting functions are referred to as the asymmetric weighting function, Type a, and the symmetric weighting function, Type b, and were originally proposed by Sakoe and Chiba.[3] Weighting function Type a weights all frames of the reference pattern equally, while weighting function Type b weights all frames of *both* the reference and the test equally. For initialization purposes, $i(0)$ and $j(0)$ are defined to be 0 and thus $\tilde{W}(1) = 1$ for weighting function Type a and $\tilde{W}(1) = 2$ for weighting function Type b.

The choice of $\mathbf{N}(\tilde{W})$ is typically made such that $D(i(k), j(k))$ is the average local distance along the path defined by $i(k)$ and $j(k)$, and is independent of both the lengths of the reference and test patterns, as well as the length of the time alignment path itself. The natural choice for $\mathbf{N}(\tilde{W})$ is thus

$$\mathbf{N}(\tilde{W}) = \sum_{k=1}^{K} \tilde{W}(k). \tag{4}$$

For weighting functions Types a and b the normalization is given by

$$\mathbf{N}(\tilde{W}_a) = \sum_{k=1}^{K} (i(k) - i(k-1)) = i(K) - i(0) = N, \tag{5a}$$

$$\begin{aligned}\mathbf{N}(\tilde{W}_b) &= \sum_{k=1}^{K} (i(k) - i(k-1) + j(k) - j(k-1)) \\ &= i(K) - i(0) + j(K) - j(0) = N + M.\end{aligned} \tag{5b}$$

Given a weighting function and a set of local constraints it is possible to define the optimal time-alignment path as that path which minimizes the total distance $D(i(k), j(k))$. More formally, if we denote the distance associated with the optimal path as $\hat{D}$, then

$$\hat{D} = \min_{K, i(k), j(k)} [D(i(k), j(k))]. \tag{6}$$

The solution to this problem may be found by dynamic programming by use of the following optimality principle:

Local Optimality: If the best path from the grid point (1, 1) to the grid point (n, m) goes through a grid point (n', m'), then the best path

from the grid point (1, 1) to the grid point (n, m) includes, as a portion of it, the best path from the grid point (1, 1) to the grid point (n', m').

Thus, if we define $D_A(n, m)$ as the minimum total distance along any path from the grid point (1, 1) to the grid point (n, m), then $D_A(n, m)$ can be computed, recursively according to the optimality principle, as

$$D_A(n, m) = \min_{n',m'} [D_A(n', m') + \hat{d}((n', m'), (n, m))], \tag{7}$$

where $\hat{d}((n', m'), (n, m))$ is the weighted distance from the grid point (n', m') to the grid point (n, m). For example, for Type I local constraints and an asymmetric weighting function, n' and m' may take on any of the following values,

$$(n', m') \in \{(n-1, m-1), (n-1, m-2), (n-2, m-1)\} \tag{8}$$

and $\hat{d}((n', m'), (n, m))$ is given by

$$\hat{d}((n-1, m-1), (n, m)) = d(n, m), \tag{9a}$$

$$\hat{d}((n-1, m-2), (n, m)) = d(n, m), \tag{9b}$$

$$\hat{d}((n-2, m-1), (n, m)) = 2d(n, m). \tag{9c}$$

Thus the full DTW recursion for Type I local constraints and weighting function Type a is given by

$$D_A(n, m) = \min[D_A(n-1, m-1) + d(n, m), D_A(n-1, m-2) + d(n, m), D_A(n-2, m-1) + 2d(n, m)]. \tag{10}$$

Using the local optimality principle, a complete DTW algorithm is given by the algorithm

Step 1. Initialize $D_A(1, 1) = d(1, 1)\bar{W}(1)$.
Step 2. Compute $D_A(n, m)$ recursively for $1 \le n \le N, \quad 1 \le m \le M$.
Step 3. $\hat{D} = D_A(N, M)/\mathbf{N}(\bar{W})$.

This completes our review of the basic principles involved in applying dynamic programming to discrete word recognition. We will now describe the difficulties which arise when DTW algorithms are applied to connected word recognition problems and then we will show how the DTW principle can be modified for word spotting and connected word recognition applications.

2.2 Difficulties in connected word recognition

We shall assume that we are given a test pattern consisting of a sequence of connected words, spoken in a normal manner, for which the global beginning and ending points have been accurately located

and for which no further segmentation has been attempted. Given such a framework, the word spotting problem is to determine all subsections of the test pattern, if any, which match with a specified reference pattern, called the keyword. Thus, for word spotting a multiplicity of regions of the test pattern must be compared with the keyword pattern.

The connected word recognition problem, on the other hand, is to piece together reference patterns (obtained, in all our work, from isolated occurrences of words) to match the test pattern. The general approach to this problem will be the one proposed by Levinson and Rosenberg,[13] namely:

(*i*) Find the reference pattern that best fits a given section of the test pattern.

(*ii*) Use the position within the test pattern at which the best matching word ends to postulate the beginning of the following word.

(*iii*) Continue to concatenate reference patterns in this manner until the test pattern is exhausted.

Dynamic time-warping algorithms, as they have been applied to discrete word recognition applications, are not directly applicable to either the word spotting or the connected word recognition problem. There are two reasons why this is so. Figure 3 illustrates some of the problems which are encountered. In this figure we show the time

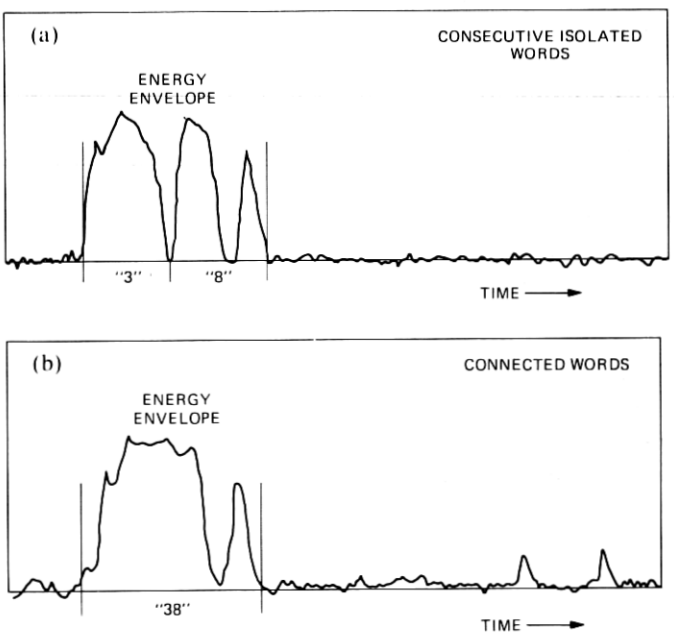

Fig. 3—Log energy for two speech utterances.

pattern for log intensity of two speech utterances, "3," "8" in part a, and "38" in part b. The utterance in part a was spoken with a discernible pause between the "3" and the "8," while the utterance in part b was spoken with no discernible pause between the "3" and the "8." Dynamic time-warping algorithms, as they have been applied to discrete word recognition, require a reliable set of word boundaries. However, as seen in Fig. 3b, a reliable segmentation for the utterance "38" is difficult, if not impossible, to obtain.

Another difficulty in using DTW algorithms, based on isolated word reference templates, for connected speech applications is the problem of coarticulation between words. For example, the final /*i*/ of the word "3" and the initial /e^i/ of the word "8" coarticulate strongly with each other. Thus, another fundamental assumption that has been relied on, namely that the characteristics of the isolated reference words which we are trying to match to our test utterance can be truly found in the test pattern, is not valid. In the next section we will describe the basic techniques that will be used to overcome these difficulties.

2.3 Basic approaches to connected speech recognition problems

In our approach to connected word recognition and word spotting we will make two changes from the structure of the isolated word DTW algorithm. One change is to no longer attempt to find the entire isolated reference pattern in the test pattern. We will still use isolated words as our reference patterns but will only expect a good match in the middle of the word, and not necessarily near the ends. Thus, we will not require that we be able to accurately match the beginning and ending points of the reference pattern to points within the test pattern. As a result, we would like to consider the possibility of overlapping reference patterns to recognize connected speech. In this manner we hope to account for both errors in the endpoint locations and for some of the gross features of coarticulation.

Another fundamental modification to the basic DTW algorithm is the use of beginning and ending *regions* rather than beginning and ending *frames*. In this manner we hope to avoid some of the problems inherent in requiring an accurate segmentation of the test utterance. Figure 4 defines, within a test pattern, a beginning region of size B (frames), with potential starting frames between b_1 and b_2 ($B = b_2 - b_1 + 1$), and an ending region of size E, with potential ending frames between e_1 and e_2 ($E = e_2 - e_1 + 1$). One possible DTW constraint would be that the best time-alignment contour may begin *anywhere* within the beginning region and end *anywhere* within the ending region. Three such potential paths are shown in Fig. 4. Such a framework would be used for word spotting, in which the beginning and ending regions correspond to the *entire* test pattern, or for connected word recogni-

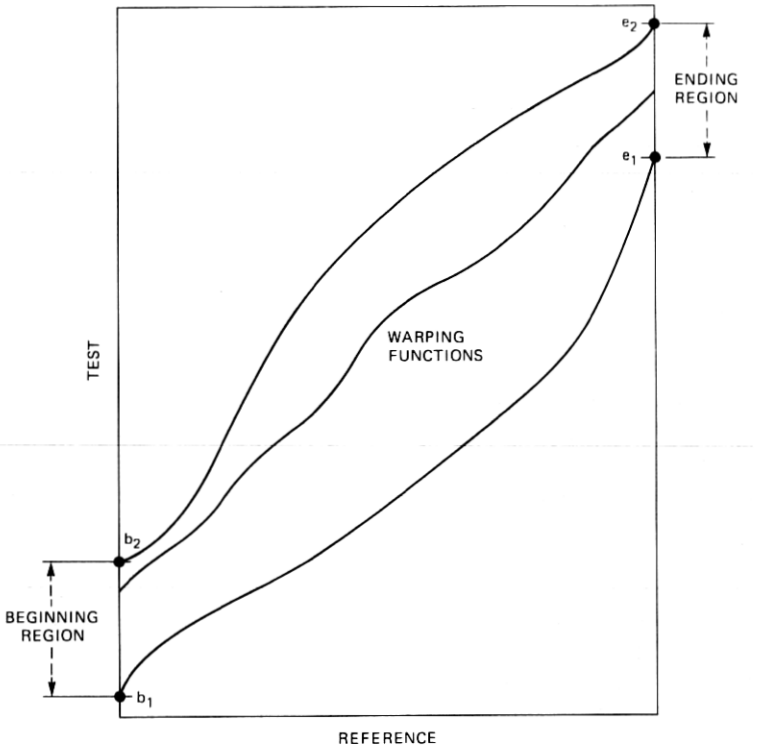

Fig. 4—Illustration of the use of beginning and ending regions.

tion, in which the ending region for one word is used to hypothesize the beginning region for the next word.

The use of beginning and ending regions modify the basic DTW algorithm by changing the constraints which are imposed on the ends of the time-alignment contour, i.e.,

$$i(1) = 1, \qquad j(1) = b, \qquad b_1 \le b \le b_2, \tag{11a}$$

$$i(K) = N, \qquad j(K) = e, \qquad e_1 \le e \le e_2. \tag{11b}$$

Thus, to find the optimal time-alignment contour, every possible beginning and ending point pair must be tried, that is,

$$\hat{D} = \min_{b_1 \le b \le b_2} \left[\min_{e_1 \le e \le e_2} \left[\min_{K, i(k), j(k)} [D(i(k), j(k)) \text{ s.t. } j(1) = b, j(K) = e] \right] \right]. \tag{12}$$

The amount of computation required to solve eq. (12) for the optimal path can be excessive, i.e., theoretically we require $B \cdot E$ separate time warps in the most general case. However, the amount of computation

required to solve eq. (12) may be reduced to a *single* time warp by judicious selection of the weighting function. If $\tilde{W}(k)$ is chosen to be the asymmetric weighting function, Type a ($\tilde{W}_a(k) = i(k) - i(k-1)$), and $\mathbf{N}(\tilde{W})$ is chosen appropriately ($\mathbf{N}(\tilde{W}_a) = N$), then $\hat{D}$ may be computed efficiently by a modified DTW algorithm as follows:

Step 1. Set $D_A(1, b) = d(1, b)$ for $b_1 \le b \le b_2$,

Step 2. Compute $D_A(n, m)$ recursively for $1 \le n \le N$, $b_1 \le m \le e_2$,

Step 3. $$\hat{D} = \frac{1}{N} \min_{e_1 \le e \le e_2} [D_A(N, e)].$$

This algorithm works because Step 1 initializes all possible beginning points, Step 2 computes the best path to a point (n, m) from any of the potential beginning points initialized in Step 1, and Step 3 finds the best possible ending point along any path from any possible beginning point. The particular choice of the asymmetric weighting function is important because its normalization factor is unaffected by the choice of the beginning or ending points, i.e., its normalization factor is always N. A dependence on the length of the test pattern, as in the symmetric weighting function, Type b, would require a separate time warp for each set of beginning and ending points because the effective length of the test $(e - b + 1)$ depends on the choice of the beginning and ending points.

An important factor, even with the savings of a single time warp, is the large amount of computation required for the DTW algorithm. Step 2 of the modified DTW algorithm is defined for $1 \le n \le N$, $b_1 \le m \le e_2$ and this region may be as large as $N \cdot M$. It is also not possible to significantly reduce this size by using restrictions on the slope of the warping contour when the ending region is left unspecified. This point is illustrated in Fig. 5, where the slope of the warping function is restricted to be between ½ and 2. We observe that, even with this restriction, when no ending region is specified, the area for which $D_A(n, m)$ must be computed is $\frac{3}{4}N^2 + B \cdot N$.

Two modifications to the DTW algorithm have been suggested to reduce this amount of computation. In particular, Sakoe and Chiba[3] have proposed that a time-warping path not be allowed to deviate significantly from a straight line, i.e., for any $i(k)$, the value of $j(k)$ is restricted such that

$$|j(k) - i(k) - \bar{b} + 1| \le R, \tag{13}$$

where $\bar{b}$ is the center of the beginning region $[\bar{b} = (b_1 + b_2)/2]$ and R is the maximum deviation which is allowed. R must be chosen to at least cover the entire beginning region, i.e., $2R + 1 \ge B$. This algorithm

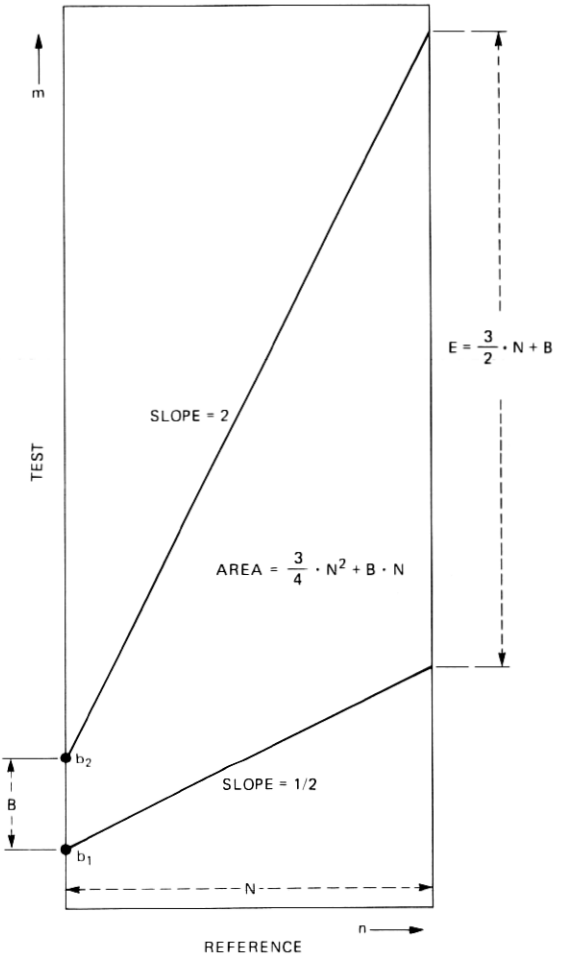

Fig. 5—Region of the (n, m) plane which is examined in a time warp for which no ending region is specified.

will be referred to as the *fixed range* DTW algorithm and is illustrated in Fig. 6a. Another range-reduction technique, proposed by Rabiner, Rosenberg, and Levinson[10] and described in detail by Rabiner and Schmidt[8] is shown in Fig. 6b. Here $j(k)$ is restricted to be within a fixed range about the best path so far, that is, the local minimum Formally, we have

$$|j(k) - c(k)| \leq \epsilon, \tag{14a}$$

$$c(k) = \underset{m}{\text{argmin}}[D_A(i(k) - 1, m)], \tag{14b}$$

$$c(1) = \tilde{b}, \tag{14c}$$

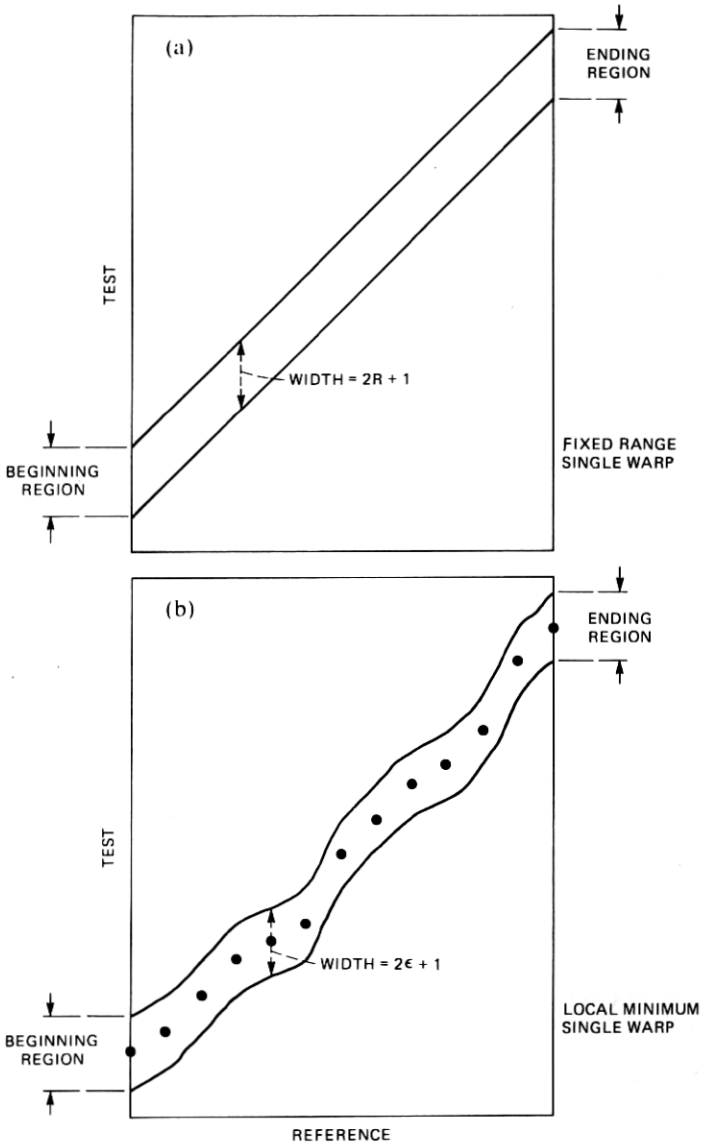

Fig. 6—Illustration of the fixed range and the local minimum DTW algorithms.

where $c(k)$ is the position, in the vertical direction, of the local minimum of $D_A(i(k) - 1, m)$, and ϵ is the allowable range about this local minimum. Thus, if $D_A(n, m)$ is computed in successive vertical strips, i.e., n is fixed and m is varied, then the range of one vertical strip is $\pm\epsilon$ about the local minimum of the previous vertical strip. This algorithm is referred to as the *local minimum* DTW algorithm.

Two fundamental differences exist between these two algorithms. The fixed range DTW algorithm, *a priori*, specifies the ending region from the specification of the beginning regions, i.e.,

$$E = 2R + 1, \tag{15a}$$

$$e_1 = \bar{b} + N - R, \tag{15b}$$

$$e_2 = \bar{b} + N + R, \tag{15c}$$

while the local minimum DTW algorithm defines the ending region implicitly from the local minimum of the last vertical strip, i.e.,

$$E = 2\epsilon + 1, \tag{16a}$$

$$e_1 = c(K) - \epsilon, \tag{16b}$$

$$e_2 = c(K) + \epsilon. \tag{16c}$$

The other fundamental difference between the two time-warping algorithms involves the number of time warps required to cover a beginning region. For the fixed range DTW algorithm the entire beginning region is most efficiently covered in a single time warp with $2R + 1 = B$, rather than several smaller time warps, because overlapping time warps may be merged together without loss of accuracy.

However, an analogous specification of the local minimum time-warping algorithm ($2\epsilon + 1 = B$) may not be truly optimal. Since one application of the local minimum DTW algorithm may follow only one local minimum path, erroneous decisions may be made because the true path may be "lost," i.e., the globally best path may not be within ϵ frames of the locally best path. As such, it may be better to try several smaller local-minimum time warps, thus allowing several different local-minimum paths to be tried, and to compare the results of these paths to determine the overall "best" path. Such a procedure is illustrated in Fig. 7. We assume that NTRY local minimum time warps are to be computed. Each time warp has (about its respective local minimum) a local range of $\pm\epsilon$ and the centers of two adjacent time warps are initially separated by δ. The entire region covered by the NTRY time warps is given by

$$\Delta = 2\epsilon + 1 + (\text{NTRY} - 1) \cdot \delta. \tag{17}$$

To cover the entire beginning region, NTRY, ϵ and δ are chosen so that $\Delta = B$.

In the next section of this paper we describe experiments designed to measure the relative strengths and weaknesses of the fixed range and the local minimum DTW algorithms and also to determine reasonable choices for the parameters δ, ϵ, and NTRY for both word spotting and connected word recognition applications.

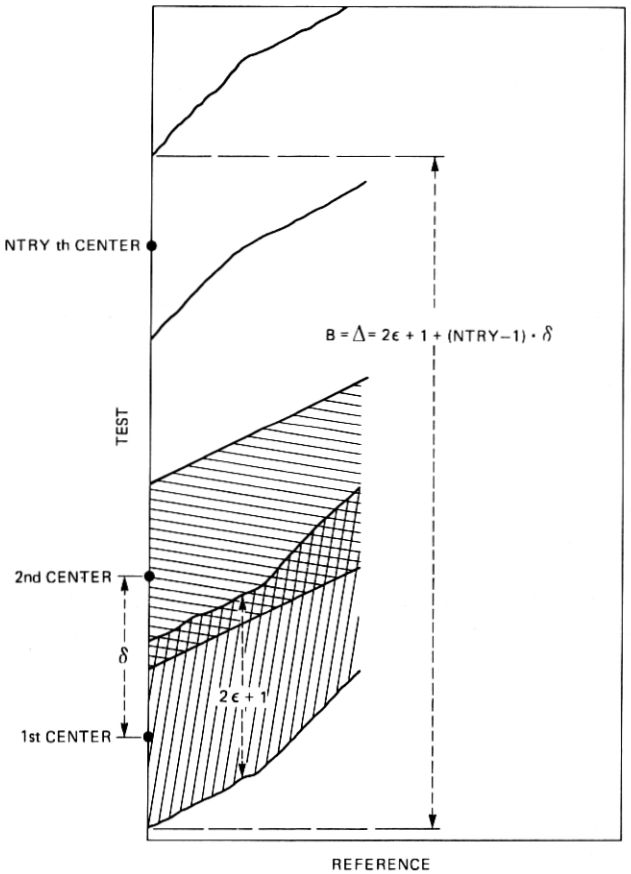

Fig. 7—Illustration of the parameters of the local minimum DTW algorithm.

III. EXPERIMENTS IN DYNAMIC TIME WARPING FOR CONNECTED SPEECH RECOGNITION

This section presents the results of experiments designed to compare the fixed range and the local minimum DTW algorithms. We also describe the results of several experiments designed to study the parameters of the local minimum algorithm. Finally, we show how these results may be applied to the problems of word spotting and connected word recognition.

3.1 Comparison of the time warping algorithms

In our initial experiment the recognition accuracies achieved by both the fixed range and the local minimum DTW algorithms for a modified *isolated* word recognition problem are compared. The test utterances consisted of 54 words from a vocabulary of computer terms,

spoken by each of 4 talkers, for a total of 216 utterances. The test utterances were recorded over a dialed-up telephone line, band-limited to 3.2 kHz, digitized at 6.67 kHz, and analyzed every 15 ms with an eighth-order LPC analysis using a 45-ms window (i.e., successive frames overlapped by 30 ms). Local distance scores, $d(i(k), j(k))$,were calculated using Itakura's log likelihood ratio.[4] The reference patterns consisted of two templates per word of the vocabulary formed by a speaker-independent clustering technique.[14]*

To evaluate the relative performance of the two DTW algorithms the test utterances were modified so that a beginning region could be specified as some range about the true beginning point. No ending region was specified. For the sake of comparison, R and ϵ were both set equal to eight frames† and NTRY was set to one. Figure 8 shows the recognition results for both algorithms as a function of the four different local constraints (used in the DTW algorithms) defined in Section 2.1. We observe that the local minimum DTW algorithm performed better than the fixed range DTW algorithm for *all* local constraints.

In another comparison we generated ten pseudo-connected test sequences by artificially embedding (at an arbitrary frame) an isolated digit into a connected-digit sequence, both uttered by the same talker. We then used both DTW algorithms to "spot" the embedded digit using two speaker-dependent templates per digit. The parameters of the two DTW algorithms that were used were the same ones as in our initial experiment ($\epsilon = 8$, $R = 8$). To spot the embedded digit, every possible beginning region of size $2\epsilon + 1$ ($= 2R + 1$) was tried. The number of times that the DTW algorithm found the (correct) best path (as determined by the lowest overall distance achieved by any beginning region) was recorded. We also recorded the ending point of the embedded word, as estimated by the word spotting procedure. Results showed that both the local minimum and the fixed range DTW algorithms were able to locate the endpoint of the embedded word with a high degree of accuracy. (The average error between the true ending frame and the estimated ending frame was 1.2 frames for both DTW algorithms.)

Figure 9 shows the relative performance of the two DTW algorithms for this simple word spotting experiment. These figures plot the number of times that the particular DTW algorithm found the proper path (as determined by the lowest-distance score achieved) for each of

---

* The speaker-independent reference template set was a subset of the 12 template per word set used in Ref. 14. This modification was used to reduce computation (and hence reduce accuracy somewhat). For the purpose of our experiments (i.e., the relative comparison of the fixed range and the local minimum DTW algorithms) this modification was of little consequence.

† Setting R and ϵ equal is a fair comparison of the two methods since the computation is the same for both methods.

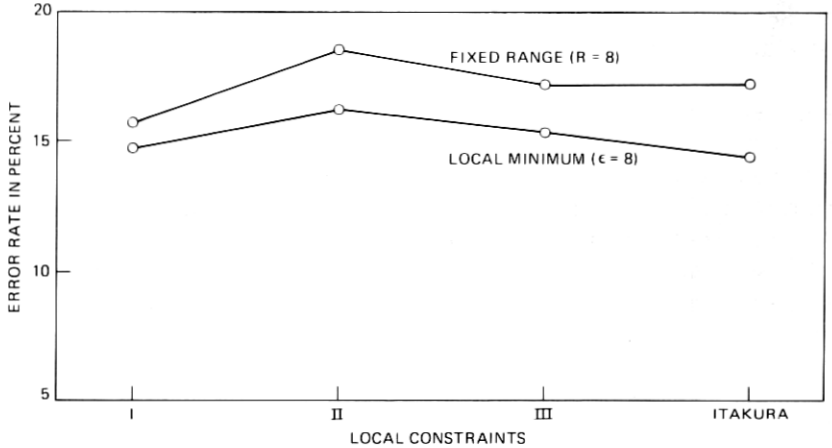

Fig. 8—Results for word recognition using both the fixed range and the local minimum DTW algorithms.

the ten embedded digits. We observe from Fig. 9 that the local minimum DTW algorithm found the best path more often than the fixed range DTW algorithm for almost all digits.

We also observe that the local minimum algorithm was able to find the best path 17 times (the maximum number possible, $2\epsilon + 1$) for 8 of the 10 digits, while the fixed range algorithm never achieved this accuracy.

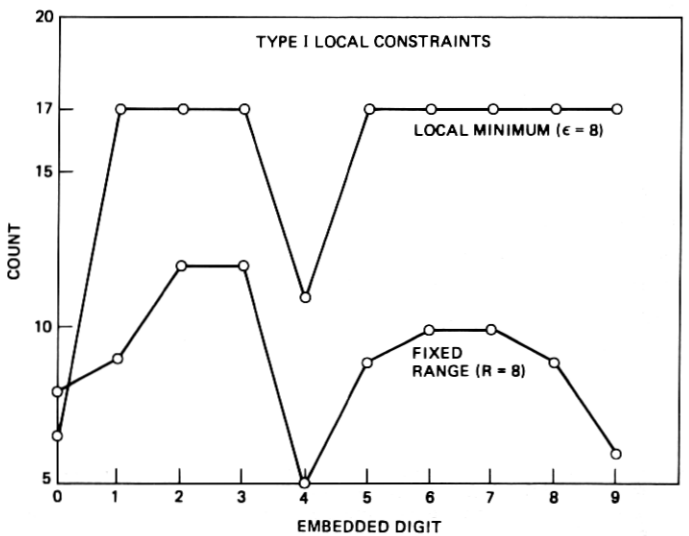

Fig. 9—Results for word spotting using both the fixed range and the local minimum DTW algorithms.

The results of these two sample experiments showed that the local minimum DTW algorithm performed consistently better than the fixed-range DTW algorithm. In the next section we describe experiments designed to more fully study some of the parameters of the local-minimum time-warping algorithm.

3.2 *Examination of the parameters of the local-minimum dynamic time-warping algorithm*

To understand the effects of the various combinations of the parameters Δ, δ, NTRY, and ϵ on the performance of the local minimum DTW algorithm, a series of connected digit-recognition experiments was performed. A total of 80 strings of from 2 to 5 connected digits each (20 strings of each length) were recorded by each of the two talkers. These strings were the same as those used by Rabiner and Schmidt.[8] In the recognition task we used two speaker-dependent templates per digit. The first step in the experiment was to "spot" the ending point of the first digit in each string via a local-minimum algorithm ($\epsilon = 11$, NTRY = 1) using the known beginning point of the first digit. Then an attempt was made to recognize the second digit in the string. Because of inaccuracies in "spotting" the ending point of the first digit, and because of coarticulation effects, it was not possible to precisely determine the beginning point of the second digit, and, as such, a beginning region for the second digit was centered around the ending frame of the first digit, as determined by the "spotting" procedure. The best candidate for the second digit was chosen as that template which achieved the lowest overall average distance, regardless of where it ended. Several values of ϵ, δ, Δ, and NTRY were used and the accuracies and distance scores for the recognition of the second digit were recorded.

Figure 10 shows, for a large value of Δ (27 in this case), the average best distance score for all NTRY time warps as a function of δ, for several values of ϵ. Two curves are shown in each part of the figure. The solid curve is the case when the reference word is the same as the second word in the test strings. The dashed curve represents the case in which the reference is different from the second word in the test string. Examination of Fig. 10 shows that the average best distance for both "same words" and "different words" increases as δ increases. However, we observe that when the reference is different from the second digit in the test utterance (i.e., the dashed curves), the average distance generally increases as δ increases, but, when the reference and the test words are the same (i.e., the solid curves), the average best distance is constant for small values of δ and increases only beyond the critical value $\delta = 2\epsilon + 1$. This critical value, $\delta = 2\epsilon + 1$ (shown by a caret in the scales of Fig. 10), is a particularly important value of δ

because for $\delta < 2\epsilon + 1$, consecutive time warps overlap in their beginning regions, and for $\delta > 2\epsilon + 1$ there are frames between two consecutive time warps which are not covered by either beginning region. When $\delta = 2\epsilon + 1$, we have the case where there is no overlap in adjacent beginning regions and no skipped frames between these regions. From the results shown in Fig. 10 we conclude that, on average, there is no loss in performance in the local-minimum DTW algorithm as long as no potential beginning frames are skipped, i.e., as long as $\delta \leq 2\epsilon + 1$.

One explanation of why δ may be taken as large as $2\epsilon + 1$, i.e., no overlapping of beginning regions, without an appreciable loss of accuracy, is shown in Fig. 11. Here we show the progress of a set of typical paths in which the starting regions overlap. By the nature of the local-minimum DTW algorithm, best paths from overlapping time warps tend

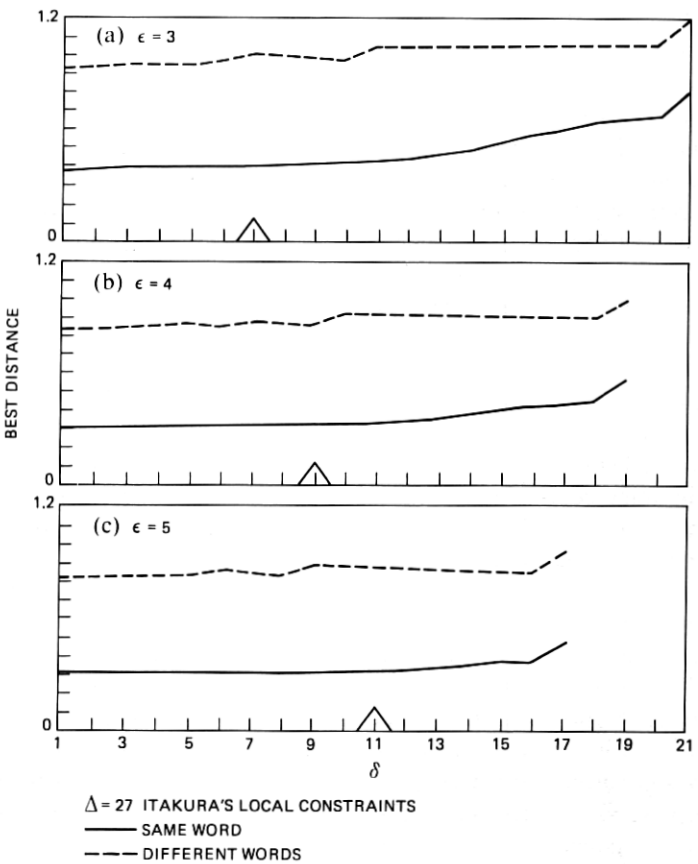

Fig. 10—Distance scores for the local minimum DTW algorithm as applied to connected digit recognition.

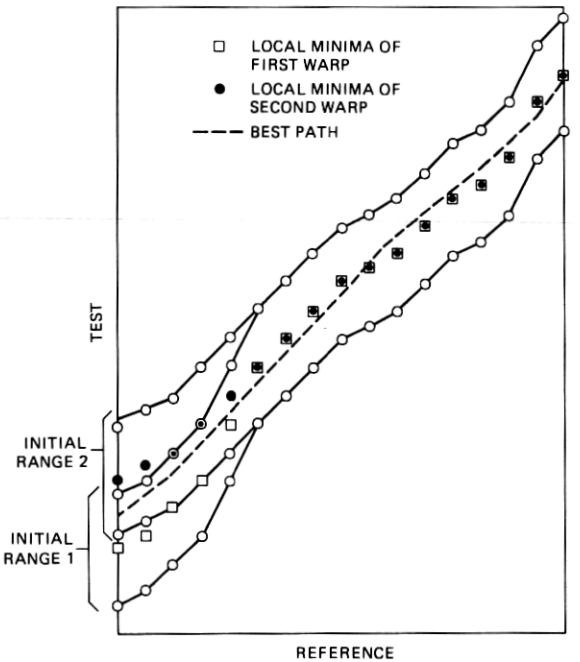

Fig. 11—Illustration of path merging for two adjacent local-minimum time warps.

to merge if there is a good path common to both of their beginning regions. Figure 12 shows the effects of path merging (of the local minimum DTW algorithm) on the digit recognition accuracies. Here we plot the recognition error rate for the second digit in the test sequences as a function of δ for various values of ϵ. We see that, for a fixed ϵ, it is possible to increase δ with essentially no loss in accuracy as long as $\delta \leq 2\epsilon + 1$.*

Figure 12 also shows that $\epsilon = 6$ provides the minimum error rate. It is reasonable to expect that as ϵ is made too small, good paths may easily become lost; but as ϵ is made too large, incorrect paths may start to generate low scores and thus cause errors. Thus, a finite value of ϵ is probably optimum. Unfortunately, such a value will have to be determined for each application.

Another interesting effect on recognition accuracy for various combinations of ϵ, δ, Δ, and NTRY is shown in Fig. 13. Here we plot recognition error rates for the second digit of our test utterances for two cases, namely $\epsilon = (\Delta - 1)/2$ (NTRY = 1), and for the best combination of ϵ, δ, and NTRY (as determined by the lowest-recog-

* Note that for Δ fixed, the largest possible δ is $\delta = \Delta - 2\epsilon - 1$ (NTRY = 2) so that the curves for the various values of ϵ in Figure 12 are defined only for those values of δ such that $\delta \leq \Delta - 2\epsilon - 1$.

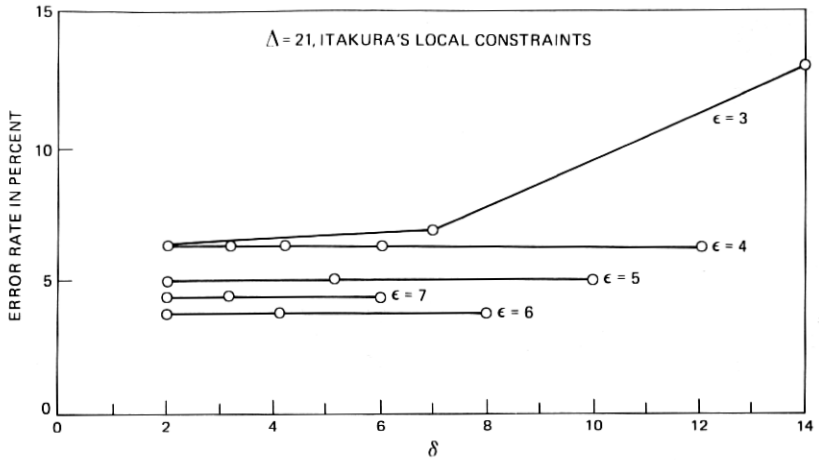

Fig. 12—Digit error rate for connected digit recognition using the local minimum DTW algorithm and several values of ϵ and δ.

nition error rate). We see that, for smaller values of Δ, a single warp performs as well as any combination of ϵ, δ, and NTRY, and as Δ increases, the differences in error rates between the best possible ϵ, δ, and NTRY combination and a single warp remains less than 2.5%. Thus, it might be possible to perform some type of connected word recognition using only a single local-minimum time warp per word. In the next section we describe how the results of our experiments have actually been applied to both word spotting and connected word recognition applications.

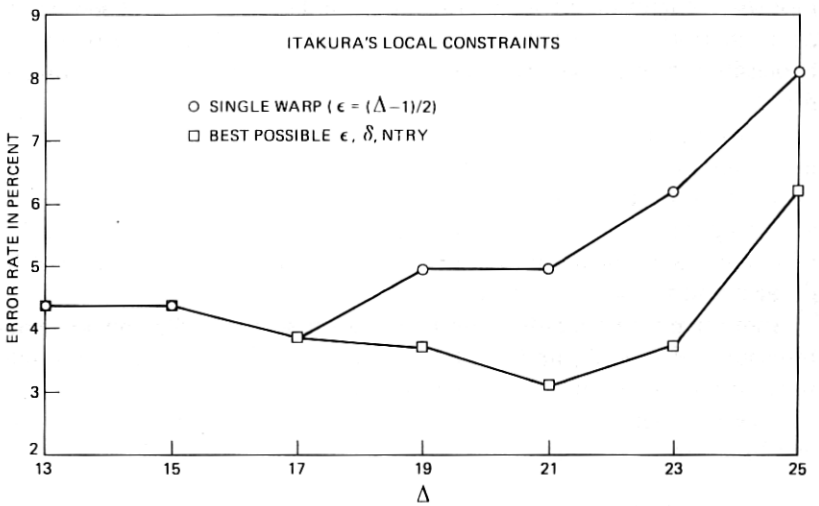

Fig. 13—Digit error rates for connected digit recognition using the local minimum DTW algorithm.

3.3 Application of DTW algorithms to word spotting and connected word recognition

We have shown that, both for connected word recognition and word spotting applications, the local minimum DTW algorithm performs consistently better than the fixed range DTW algorithm. We have also shown that, given a value of ϵ, δ may be chosen as large as $\delta = 2\epsilon + 1$ without significant degradation in the performance of the local minimum DTW algorithm. Since, for a fixed beginning region (i.e., a fixed Δ), the number of time warps is given by NTRY $= 1 + (\Delta - 2\epsilon - 1)/\delta$, the best choice for δ is $\delta = 2\epsilon + 1$. This minimizes the number of time warps which need to be performed. For the problem of word spotting the obvious choice for Δ is $\Delta = M$, i.e., the entire length of the test pattern. For this case optimal values of ϵ and NTRY must still be determined. In general, the selection of ϵ and NTRY depends on several factors. As ϵ is increased, the chance of a missed keyword decreases because more paths are examined, but the chance of a false alarm increases. Also, as ϵ increases, the value of NTRY decreases [NTRY $= \Delta/(2\epsilon + 1)$ for $\delta = 2\epsilon + 1$], thereby reducing the amount of computation required. Thus, misses, false alarms, and the amount of computation must be traded-off in the selection of ϵ and NTRY for a word spotting application.

In a connected word recognition application, however, we not only must choose ϵ and NTRY but must also choose Δ. We have shown that for $\Delta \leq 17$ frames, it is possible to do connected digit recognition using only a single local-minimum time warp per word. However, we also found that the best recognition accuracy was achieved with $\Delta = 21$ but not with a single local-minimum time warp. Thus, there is an apparent trade-off between recognition accuracy and speed of computation. However, work by Rabiner and Schmidt[8] has shown that it is better not to center the beginning region of one word around the end of the previous word, as we did, but, rather, to center the beginning regions of one word several frames earlier than the ending region of the previous word. The reason for this is that the isolated reference patterns tend to be longer than the spoken connected words, and thus, the time warps tend to overestimate the ending frame of each word. We tried a simple experiment in which the beginning region of one word was centered eight frames earlier in the test pattern than the end of the previous word. The values of ϵ, NTRY, and Δ were $\epsilon = 8$, NTRY $= 1$, and $\Delta = 17$. Using these values and the same test utterances used by Rabiner and Schmidt,[8] i.e., 80 sequences of from 2 to 5 digits each spoken once by each of six talkers, we achieved a string recognition rate of 429 correct strings out of 480 possible. This may be compared with a total of 442 correct strings using $\epsilon = 8$, $\delta = 3$, and NTRY $= 4$, as reported by Rabiner and Schmidt. It should be noted,

however, that the system of Rabiner and Schmidt used multiple candidate strings while our simple experiment did not. When we reran the system of Rabiner and Schmidt using only a single candidate string ($\epsilon = 8$, $\delta = 3$, NTRY = 4) we found only 430 correct strings out of the 480 possible. Thus, with a single local-minimum time warp per word we achieved results comparable to those achieved by the use of four local-minimum time warps per word.

IV. CONCLUSIONS

We have shown that dynamic time warping algorithms can be efficiently applied to both word spotting and connected word recognition. We have demonstrated the relative performance superiority of the local minimum DTW algorithm over the fixed-range DTW algorithm. It was also shown that the beginning regions of successive applications of the local minimum DTW algorithm need not overlap to achieve accuracy comparable to overlapping beginning regions. We have found that, for small beginning regions (small Δ), a single local-minimum time warp [with $\epsilon = (\Delta - 1)/2$, NTRY = 1] was as accurate as (and more computationally efficient than) any combination of the parameters ϵ, δ, and NTRY. Finally, we found that an extremely simple connected digit recognition system, i.e., a single local-minimum time warp per word using only one candidate string, achieved a string recognition rate of nearly 90 percent.

REFERENCES

1. T. B. Martin, "Practical Applications of Voice Input to Machines," Proc. IEEE, *64* (April 1976), pp. 487–501.
2. G. M. White and R. B. Neely, "Speech Recognition Experiments with Linear Prediction, Bandpass Filtering and Dynamic Programming," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-24* (April 1976), pp. 183–8.
3. H. Sakoe and S. Chiba, "A Dynamic Programming Approach to Continuous Speech Recognition," Proc. Int. Congress Acoustics, Budapest, Hungary, 1971, Paper 20C-13.
4. F. Itakura, "Minimum Prediction Residual Principle Applied to Speech Recognition," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-23* (February 1975), pp. 67–72.
5. J. S. Bridle, "An Efficient Elastic Template Method for Detecting Given Words in Running Speech," Proc. British Acoust. Soc. Meetings, London, England, April 1973, Paper 73SHC3.
6. R. W. Christiansen and C. K. Rushforth, "Detecting and Locating Keywords in Continuous Speech Using Linear Preductive Coding," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-25* (October 1977), pp. 361–7.
7. H. Sakoe, "Two-Level DP Matching—A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-27* (December 1979), pp. 588–95.
8. L. R. Rabiner and C. E. Schmidt, "Application of Dynamic Time Warping to Connected Digit Recognition," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-28* (August 1980).
9. C. S. Myers and L. R. Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," IEEE Trans. Acoust. Speech, Signal Proc., to appear.
10. H. Sakoe and S. Chiba, "Dynamic Programming Optimization for Spoken Word

Recognition," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-26* (February 1978), pp. 43–9.
11. L. R. Rabiner, A. E. Rosenberg, and S. E. Levinson, "Considerations in Dynamic Time Warping for Discrete Word Recognition," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-26* (December 1978), pp. 575–82.
12. C. S. Myers, L. R. Rabiner, and A. E. Rosenberg, "Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition," IEEE Trans. Acoust. Speech, Signal Proc., to appear.
13. S. E. Levinson and A. E. Rosenberg, "A New System for Continuous Speech Recognition—Preliminary Results," Proc. Int. Conf. Acoust. Speech, Signal Proc. (April 1979), pp. 239–44.
14. L. R. Rabiner and J. G. Wilpon, "Speaker Independent, Isolated Word Recognition for a Moderate Size (54 Word) Vocabulary," IEEE Trans. Acoust. Speech, Signal Proc., *ASSP-27* (December 1979), Part I, pp. 583–7.