STUDIA PHONOLOGICA XX (1986)

Connected Spoken Digit Recognition by Augmented Continuous DP Matching and its Evaluation

Sei-ichi NAKAGAWA and Tatsuzi ITOH

ABSTRACT

Recently, we proposed the Augmented Continuous dynamic time warping algorithm for connected spoken word recognition. The algorithm is based on the same principle as the Two Level DP and Level Building DP. Our algorithm obtains a near optimal solution for the recognition principle based on pattern matching. However, it is computationally more efficient than the conventional methods and does not require much memory storages.

In this paper, we modified this algorithm on the following points: 1) matching methods for a silence in speech, 2) scoring mechanisms for connected words, 3) simple word spotting algorithms, 4) generation methods of the best plural candidates and 5) endpoint-free matching mechanism. We apply this modified algorithm to connected spoken digit recognition and evaluate it.

We find from experiments as follows: First, the consideration of speech silence improves the recognition performance. Secondly, the matching result of longer reference is more important than that of shorter reference in the unknown case of number of words in a test sample. Thirdly, the recognition results by simple algorithms for detecting two word spotting candidates are comparable with the results by the original complex algorithm. Forthly, the optimal strategy for the generation of the best plural candidates of word sequences is better than the suboptimal strategy. Fifthly, the endpoint-free matching mechanism improved the recognition rate.

By using one reference pattern per digit, we obtain the string recognition rate of 89.0% with the top choice and 95.3% with the top 3 choices in a speaker-dependent mode. Finally, we compare this algorithm with traditional algorithms such as the Two Level DP and find that this algorithm has almost the same performance as the traditional algorithms.

I. INTRODUCTION

The technique of dynamic time warping (DTW) by using dynamic programm-

Sei-ichi NAKAGAWA (中川聖一): Associate Professor, Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, 440 Japan.

Tatsuzi ITOH (伊藤立治): Graduate Student, Department of Information and Computer Sciences, Toyohashi University of Technology.

The authors have done the research under the direction of Dr. Toshiyuki Sakai, Professor of Information Science, Kyoto University.

ing is a powerful tool for isolated spoken word recognition [1, 2]. Vintsyuk proposed a connected spoken word recognition method based on this principle in 1970 [3]. In 1975, Sakoe also proposed independently another algorithm from the view point of pattern matching [called Two Level DP matching (TLDP)] [4].

In 1981, Myers and Rabiner presented a new DTW-based connected word recognition algorithm, called Level-Building DP matching (LB) [5]. This algorithm produces the same result as TLDP except for a difference in the matching window. This requires less computation, but is not a real-time-oriented algorithm. Therefore, Sakoe and Watari constructed a new real-time-oriented algorithm by changing the computation order of LB; they called it clockwise DP matching (CWDP) [6]. It uses far less computation, but more memory storages.

Recently, one of the authors proposed three new algorithms based on the same principle as described above [7]. One is the Constant Time Delay DP matcfiing (CTDP). This algorithm is an extension of TLDP; namely, it executes the computation after a constant time delay (every W frames) instead of every frame. This decreases computation of local distances between frames by a factor of 4 to 10 while preserving a real-time-oriented algorithm. This can be easily modified to the variable constant time delay DP matching and becomes more efficiently.

The second is the O(n) DP matching. This is able to recognize connected spoken words by one level (pass) DP matching for every reference pattern. The total amount of computation is the same as that of isolated spoken word recognition by DP matching; that is, a factor of R (the width of the matching window) smaller than for TLDP. Bridle et al. also proposed independently an almost identical method to the O(n) DP and a finite-state-automaton-controlled CWDP [8]. Although their algorithm was not described clearly [called one-pass DP], recently, Ney described the same algorithm clearly [called one-stage DP] [9].

The last is the Augmented Continuous DP matching based on word spotting [10]. Although this algorithm obtains only an approximate solution for the recognition principle based on pattern matching, it requires less computation than the others. Therefore it is useful for connected word recognition with syntactical constraints in a large vocabulary.

In this paper, we modify this algorithm on the following points: 1) matching methods for a silence in speech, 2) scoring mechanisms for connected words, 3) simple word spotting algorithms, 4) generation methods of the best plural candidates and 5) endpoint-free matching mechanism. We apply this modified algorithm to connected spoken digit recognition. Finally, we compare this algorithm with traditional algorithms such as the Two Level DP and evaluate it.

II. Formalization of Connected Spoken Word Recognition by Pattern Matching [4]

In this section, we describe the formalization of connected spoken word recogni-

tion by pattern matching. This formula was proposed by Sakoe [4] and solved by the Two Level DP matching algorithm. Our Augmented Continuous DP matching algorithm is also based on this formalism. The detailed description is shown in our companion papers [7] [10].

2.1. Notation

n: n-th word.

N: Vocabulary size (number of references)

X: Number of words in a test pattern.

J^n: Length (in frames) of the reference pattern for the n-th word.

R^n: Reference pattern for the n-th word:

$$R^n=b_1{}^n b_2{}^n \ldots b_{J^n}{}^n,$$

where $b_j{}^n$ is the feature vector of the j-th frame.

I: Length of a test pattern:

$$T=a_1 a_2 \ldots a_I,$$

where a_i is the feature vector of the i-th frame.

$d^n(i.j)$: Local distance between a_i and $b_j{}^n$.

$D_X(i)$: Minimum cumulative distance when $a_1 a_2 \ldots a_i$ is matched with any concatenation of x reference patterns.

$N_X(i)$: Last reference word in the concatenation of references to satisfy $D_X(i)$.

$B_X(i)$: Beginning point in the test pattern for $N_X(i)$ minus 1.

$D(i)$: Minimum cumulative distance when $a_1 a_2 \ldots a_i$ is matched with any concatenation of various reference patterns.

$N(i)$: Last reference word in the concatenation of references to satisfy $D(i)$.

$B(i)$: Beginning point in the test pattern for $N(i)$ minus 1.

$D^n(m:i)$: Minimum cumulative distance between $a_m a_{m+1} \ldots a_i$ and the n-th reference pattern $b_1{}^n b_2{}^n \ldots b_{J^n}{}^n$.

$D_X{}^n(i,j)$: $\min_m \{D_{X-1}{}^n(m)+$(minimum cumulative distance between $a_{m+1} \ldots a_i$ and $b_1{}^n b_2{}^n \ldots b_j{}^n$)$\}$.

$B_X{}^n(i,j)$: Beginning point in the test pattern for the n-th word corresponding to $D_X{}^n(i,j)$ minus 1.

2.2 Formalization of Connected Word Recognition Principle

Let us consider the connected spoken word recognition problem of finding the sequence of reference patterns, $\hat{R}=R^{n(1)}R^{n(2)} \ldots R^{n(X)}$ of length x which best matches minimum cumulative distance) the test pattern T, over all possible concatenations of x references. If the number of words in the test pattern is not known, the minimization should be performed over all lengths x.

The basic procedure for finding $\hat{R}$ is to solve a time alignment problem between T and R using a dynamic time warping method (DP matching). Therefore we should use an asymmetric DP path to apply dynamic programming to connected word recognition [4]. Figure 1 shows various types of asymmetric DP paths [10].

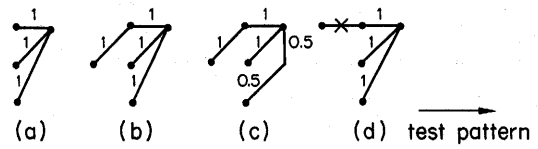

Fig. 1. Examples of asymmetric DP path and weight (base axis: test pattern)

In this paper, we use the type (a) for simplicity in explaining the algorithm.

The principle described above is defined as follows: Let $d^R(i, j)$ and $u(i)$ represent the local distance between a_i and the j-th frame in R (concatenation of references) and a time-warping function, respectively. The optimal concatenation of references is

$$\hat{R}=\underset{R}{\operatorname{argmin}}\ D(T,\ R),$$

where
$$D(T, R)=\min_{u(i)} \sum_{i=1}^{I} d^R(i, u(i)) \tag{1}$$

with $R=b_1{}^{n(1)}b_1{}^{n(1)}\ldots b_{J_{n(1)}}{}^{n(1)}b_1{}^{n(2)}b_2{}^{n(2)}\ldots b_{J_{n(2)}}{}^{n(2)}\ldots b_1{}^{n(X)}b_2{}^{n(X)}\ldots b_{J_{n(X)}}{}^{n(X)}=b_1{}^Rb_2{}^R\ldots b_{J^R}{}^R$, $0\leq u(i)-u(i-1)\leq 2$, $u(1)=1$, and $u(I)=J^R$.

2.3 Recursive Formula for Solution

Case (i): Number of words in a test pattern known.

The problem is solved by the following DP equation:

$$D_X(0)=0,\ B_X(0)=0 \text{ for } x=1, 2, \ldots, X$$
$$D_X(i)=\min_{n,m}\{D_{X-1}(m)+D^n(m+1:i)\}=\min_n D_X{}^n(i, J^n) \text{ for } x=1, 2, \ldots, X$$
$$N_X(i)=\hat{n},\ B_X(i)=\hat{m} \tag{2}$$

where $\hat{n}$ and $\hat{m}$ satisfy the equation (2). $D^n(m+1:i)$ is calculated by the following equation:

$$D^n(m+1:i)=\min_{u(k)} \sum_{k=m+1}^{i} d^n(k, u(k)), \tag{3}$$

where, $0\leq u(k)-u(k-1)\leq 2$, $u(m+1)=1$ and $u(i)=J^n$.

After repeating the DP equation (2) over $1\leq i\leq I$, the recognition result is decided in the inverse order of a word sequence by the following algorithm:

1. $i=I$, $x=X$
2. $n=N_X(i)$ ... output
3. If $B_X(i)\neq 0$, $i=B_X(i)$, $x=x-1$. Go to 2.
   If $B_X(i)=0$, stop.

Case (II): Number of words in a test pattern unknown.

The problem is solved by the following DP equation:

$$D(0)=0,\ B(0)=0,$$
$$D(i)=\min_{n,m,x}\{D_X(m)+D^n(m+1:i)\}$$
$$=\min_{n,m}\{D(m)+D^n(m+1:i)\}=\min_n D^n(i, J^n)$$
$$N(i)=\hat{n},\ B(i)=\hat{m} \tag{4}$$

where $\hat{n}$ and $\hat{m}$ satisfy the equation (4). After repeating the DP equation (4) over $1\leq i\leq I$, the recognition result is decided in the inverse order of a word sequence by

the following algorithm:

1. $i=I$
2. $n=N(i)$ ... output
3. If $B(i)\neq 0$, $i=B(i)$. Go to 2.
   If $B(i)=0$, stop

III Augmented Continuous DP Matching Algorithm [7] [10]

In this section, we describe in brief the original Augmented Continuous DP matching algorithm. This algorithm will be modified in the following sections.

3.1 Notation

$\bar{D}^n(i,j)$: $\min_m$ (minimum cumulative distance between $a_m a_{m+1}\ldots a_i$ and $b_1{}^n b_2{}^n \ldots b_j{}^n$), where the basic axis for an asymmetric DP path is the reference axis.

$\bar{B}^n(i,j)$: argmin_m (minimum cumulative distance between $a_m a_{m+1}\ldots a_i$ and $b_1{}^n b_2{}^n \ldots b_j{}^n$).

$\bar{D}^n(i,j,k)$: k-th $\min_m$ (minimum cumulative distance between $a_m a_{m+1}\ldots a_i$ and $b_1{}^n b_2{}^n \ldots b_j{}^n$).

$\bar{B}^n(i,j,k)$: arg k-th $\min_m$ (minimum cumulative distance between $a_m a_{m+1}\ldots a_i$ and $b_1{}^n b_2{}^n \ldots b_j{}^n$), where arg k-th $\min_m f(m)$ means $\hat{m}$ such that $f(\hat{m})$ is the k-th minimum of $f(m)$ over all m.

Let $v(k)$ and Figure 2(a) be a time-warping function and DP path, respectively [10]. $\bar{D}^n(i,j)$ is defined as follows:

$$\bar{D}^n(i,j)=\min_{v(k)} \sum_{k=1}^{j} d^n(v(k),k), \tag{5}$$

where $0\leq v(k)-v(k-1)\leq 2$, $1\leq v(1)\leq i$ and $v(j)=i$. We have

$$\bar{D}^n(i,j)=\bar{D}^n(i,j,1),\quad \bar{B}^n(i,j)=\bar{B}^n(i,j,1),$$
$$\bar{D}^n(i,j,1)\leq \bar{D}^n(i,j,2)\ldots\leq \bar{D}^n(i,j,K),$$
$$\bar{B}^n(i,j,k)\neq \bar{B}^n(i,j,h) \text{ for } k\neq h.$$

3.2 Word Spotting Algorithm

For speech understanding systems, the problem of detecting and locating a specific word in continuous speech has been considered by using a nonlinear time-warping procedure (DP matching) [10, 11]. This problem is referred to as the word spotting problem.

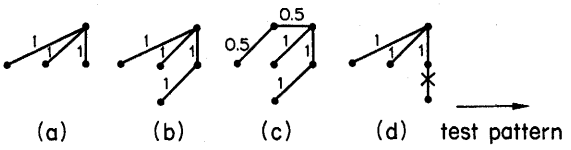

Fig. 2. Examples of asymmetric DP path and weight (base axis: reference pattern)

If we could calculated $\bar{D}^n(i, J^n)$ for the i-th frame in the test pattern and the n-th reference pattern, the word spotting problem would be solved, that is, if $\bar{D}^n(i, J^n)$ satisfies the threshold for word detection, the location can be regarded as $\bar{B}^n(i, J^n) \sim i$ in the test pattern.

$\bar{D}^n(i, J^n)$ and $\bar{B}^n(i, J^n)$ are calculated as the following DP equation:

1. Initialize
   $\bar{D}^n(-1, j)=\bar{D}^n(0, j)=\infty$ for $n=1, 2, \ldots, N$; $j=1, 2, \ldots, J^n$.
2. Execute steps 3, 4, 5 for $i=1, 2, \ldots, I$.
3. Execute steps 4, 5 for $n=1, 2, \ldots, N$.
4. $\bar{D}^n(i, 1)=d^n(i, 1)$,
   $\bar{B}^n(i, 1)=i$.
5. For $j=2, 3, \ldots, J^n$.
   $\hat{\imath}=\underset{i-2\leq i'\leq i}{\operatorname{argmin}} \bar{D}^n(i', j-1)+d^n(i, j)$
   $\bar{D}^n(i, j)=\bar{D}^n(\hat{\imath}, j-1)+d^n(i, j)$
   $\bar{B}^n(i, j)=\bar{B}^n(\hat{\imath}, j-1)$.

3.3 Application to Connected Word Recognition

We proposed a new connected word recognition algorithm based on our word spotting algorithm [7] [10]. Our method is based on the formalization of equations (2) and (4), but gives an approximate solution for them.

In equations (2) and (4), we must calculate $D^n(m+1 : i)$ for $m=0, 1, 2, \ldots, i-1$. This cumulative distance depends on the matching length in the test pattern, that is, $i-m$. On the other hand, $\bar{D}^n(i, J^n)$ depends on the length of reference, that is, J^n. Therefore we can obtain $D^n(m+1 : i)$ approximately from $\bar{D}^n(i, J^n)$ as follows:

$$D^n(m+1 : i) \leftarrow \bar{D}^n(i, J^n) \times (i-m)/J^n \qquad (6)$$
$$m+1=\bar{B}^n(i, J^n).$$

Although the location of $\bar{B}^n(i, J^n) \sim i$ in the test pattern is the most reliable candidate location for the reference and i-th frame, m is fixed at $\bar{B}^n(i, J^n)-1$ in this modification. Therefore we try to estimate $D^n(m+1 : i)$ in some locations as follows:

$$\begin{aligned}
D^n(m-r_1+1 : i) &\leftarrow \bar{D}^n(i, J^n) \times (i-m+r_1)/J^n. \\
&\vdots \\
D^n(m+1 : i) &\leftarrow \bar{D}^n(i, J^n) \times (i-m)/J^n. \\
&\vdots \\
D^n(m+r_2+1 : i) &\leftarrow \bar{D}^n(i, J^n) \times (i-m-r_2)/J^n. \\
m+1 &= \bar{B}^n(i, J^n).
\end{aligned}$$

where r_1 and r_2 delimit the estimation range. If we want to calculate $D^n(m+1 : i)$ for more locations, we should modify our word spotting algorithm as the following algorithm.

4. For $k=1, 2, \ldots, K$.
   $\bar{D}^n(i, 1, k)=d^n(i, 1)$.
   $\bar{B}^n(i, 1, k)=i$.
5. Execute 5') for $j=2, 3, \ldots, J^n$.

5'. For $k=1, 2, \ldots, K$,

$\hat{\imath}, \hat{k}=\underset{i-2\leq i'\leq i,}{\arg} \; k\text{-th} \min_{1\leq k'\leq K} \bar{D}^n(i', j-1, k')$.

$\bar{D}^n(i,j,k)=\bar{D}^n(\hat{\imath}, j-1, \hat{k})+d^n(i,j)$,

$\bar{B}^n(i,j,k)=\bar{B}^n(\hat{\imath}, j-1, \hat{k})$,

where the following condition should be satisfied:

$\bar{B}^n(i, j, k)\neq\bar{B}^n(i, j, h)$ for $1\leq h\leq k$.

Thus $D^n(m+1:i)$ is obtained from:

6. For $k=1, 2, \ldots, K$,

$m=\bar{B}^n(i, J^n, k)-1$

$D^n(m-r_1+1:i) \leftarrow \bar{D}^n(i, J^n, k)\times(i-m+r_1)/J^n$,

$\vdots$

$D^n(m+1:i) \leftarrow \bar{D}^n(i, J^n, k)\times(i-m)/J^n$,

$\vdots$

$D^n(m+r_2+1:i) \leftarrow \bar{D}^n(i, J^n, k)\times(i-m-r_2)/J^n$.

3.4 A Connected Spoken Word Recognition Algorithm by Augmented Continuous DP

In this subsection, we present a connected spoken word recognition algorithm by Augmented Continuous DP in the known case of number of words in a test pattern [10]. In the companion paper [7], we described a connected spoken word recognition algorithm with syntactic constraints by the algorithm.

1. Initialize

$D_0(0)=0 \quad B_0(0)=0$

$\bar{D}^n(-1, j, k)=\bar{D}^n(0, j, k)=\infty$ for $n=1, 2, \ldots, N$; $j=1, 2, \ldots, J^n$; $k=1, 2, \ldots, K$.

2. Execute steps 3—10 for $i=1, 2, \ldots, I$.
3. Execute steps 4—8 for $n=1, 2, \ldots, N$.
4. $\bar{D}^n(i, 1, k)=d^n(i, 1)$, $\bar{B}^n(i, 1, k)=i$ for $k=1, 2, \ldots, K$.
5. Execute step 6 for $j=2, 3, \ldots, J^n$.
6. For $k=1, 2, \ldots, K$,

$\hat{\imath}, \hat{k}=\underset{i-2\leq i'\leq i,}{\arg} \; k\text{-th} \min_{1\leq k'\leq k} \bar{D}^n(i', j-1, k')$,

$\bar{D}^n(i, j, k)=\bar{D}^n(\hat{\imath}, j-1, \hat{k})+d^n(i, j)$

$\bar{B}^n(i, j, k)=\bar{B}^n(\hat{\imath}, j-1, \hat{k})$,

where $\bar{B}^n(i, j, k)<\bar{B}^n(i, j, h)-r_1$ or $\bar{B}^n(i, j, k)>\bar{B}^n(i, j, h)+r_2$ for $1\leq h\leq k$.

7. Execute step 8 for $k=1, 2, \ldots, K$.
8. For $r=-r_1, -r_1+1, \ldots, -1, 0, 1, 2, \ldots, r_2$,

$m=\bar{B}^n(i, J^n, k)-1$,

$D^n(m+r+1:i)=\bar{D}^n(i, J^n, k)\times(i-m-r)/J^n$.

9. Execute step 10 for $x=1, 2, \ldots, X$.
10. For $m=\min_{1\leq k\leq K} \bar{B}^n(i, J^n, k)-r_1-1, \ldots, \max_{1\leq k\leq K} \bar{B}^n(i, J^n, k)+r_2-1$; $n=1, 2, \ldots, N$.

$\hat{m}, \hat{n}=\underset{m,n}{\operatorname{argmin}} \{D_{X-1}(m)+D^n(m+1:i)\}$,

$D_X(i)=D_{X-1}(\hat{m})+D^{\hat{n}}(\hat{m}+1:i)$,

$N_X(i)=\hat{n}$, $B_X(i)=\hat{m}$.

11. Word decision process (refer to section 2.3).

IV. CONSIDERATION OF SILENCE IN SPEECH

4.1 Silence in a spoken word

There are short silences at the front of voiceless stops and fricatives. The information such a silence is effective for the recognition of stop consonants or words. Therefore we introduce a strategy for taking account of silence parts in a word.

On the computation of local frame distance between the i-th frame of test pattern and the j-th frame of n-th reference pattern (that is, $d^n(i,j)$), if both of them have smaller powers than the preset threshold, its local distance is defined as the constant small distance. Figure 3 illustrates the procedure.

4.2 Silence at a boundary between words

Even if we utter words in a manner of continuous mode, there may exist a silence between words. Therefore we should take into a silence at a boundary between words. One simple solution is to eliminate all silence parts in the test pattern and to shorten the test pattern. However, this approach has a disadvantage that it discards the important information such as a silence being at the front of voiceless stop or choked sound.

We propose an another strategy. Besides reference templates of vocabulary words, we introduce a reference pattern of silence. This reference consists of only one frame and is a pseudo-template. An application of pseudo-template was introduced for taking into breath noise and illegal words by Bridle et al. [8]. We modify this approach for taking into silence parts at word boundaries. It produces the constant small local distance for any test frame so long as the power of its frame is smaller than the preset threshold Otherwise, the distance is set to an infinitive value. It is allowed to match with consecutive frames of a test pattern. In other words, the DP path is only one as follows:

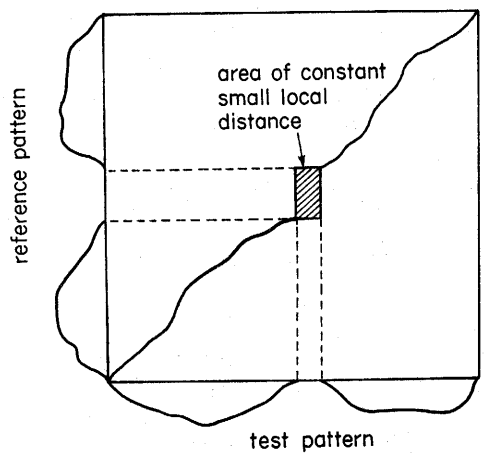

Fig. 3. Matching of silence in a spoken word.

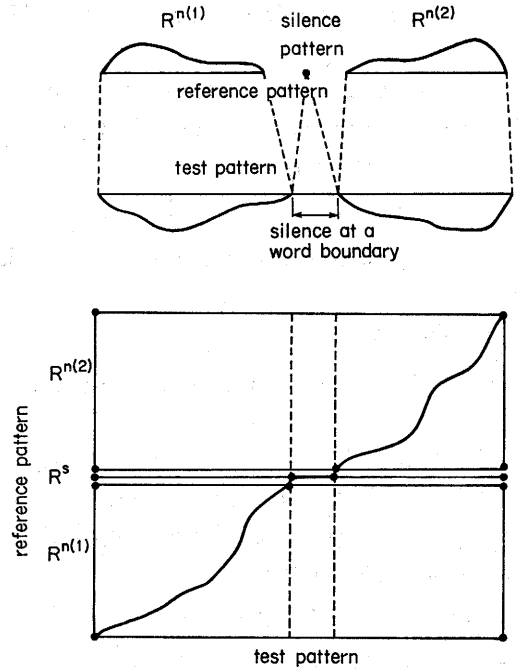

Fig. 4. Matching of silence at a boundary between words.

$m=0$,
For $i=1, 2, \ldots, I$,
$D^s(i: i-1)=0$,
$D^s(m+1: i)=D^s(m+1: i-1)+d^s(1, i)$
if $D^s(m+1: i)$ is an infinitive value. then $m=i$. (7)

Where the index s means the reference of silence ($J^s=1$) and $d^s(1, i)$ is the constant small value or an infinitive value.

Figure 4 illustrates a matching result by using this procedure, where R^s denotes the reference pattern of silence.

V. Scoring Mechanisms for Connected Words

The number of additional operations of local distance is $i-m$ for $D^n(m+1: i)$ and it is related to the length (in frameas) of the reference pattern. Therefore, the cumulative distance is very influenced by matching results of long reference patterns. In this scoring mechanism, since a short word is evaluated by the smaller weight than a long word, such a short word may be misrecognized.

In this section, we investigate other scoring mechanisms for connected words. They are based on the normalization of cumulative distance by the length of a reference pattern. Such normalization cannot be implemented by the Level Building DP or O(n) DP, but it can be implemented easily by the Two Level DP or Augmented Continuous DP.

(1) Normalization by the length of reference pattern and addition of normalized score

This scoring mechanism evaluates the matching result for all reference patterns by the same weight. This is implemented by the following:

$$D^n(m+1:i) \leftarrow \bar{D}^n(i, J^n)/J^n, \tag{8}$$
$$D_X(i)=\min_m \{D_{X-1}(m)+D^n(m+1:i)\}$$

In the unknown case of number of words in a test pattern, the number of words is estimated as follows:

$$\hat{x}=\operatorname*{argmin}_x D_X(I)/x.$$

(2) Normalization by length of reference pattern and multiplication of normalized score

In the literature [11], we tried some scoring mechanisms for a sequence of syllables uttered in isolation and obtained the best result by using the scoring mechanism by multiplication of normalized score. This is implemented by the following:

$$D^n(m+1:i) \leftarrow \bar{D}^n(i, j^n)/J^n,$$
$$D_X(i)=\min_m \{D_{X-1}(m)\times D^n(m+1:i)\}.$$

If we replace $\log D_X(i)$ to $D_X(i)$ again, the above equation becomes

$$D_X(i)=\min_m \{D_{X-1}(m)+\log D^n(m+1:i)\}. \tag{9}$$

VI. Word Spotting Algorithms for K=2

In subsection 3.3, we proposed a word spotting algorithm which calculates K candidates for each ending frame in a test pattern. However even if $K=2$, the amount of computation for cumulative distances becomes 4 to 5 times as compared with the case of $K=1$. Therefore, we propose two word spotting algorithms for $K=2$, which are less computation than the above method.

(1) Two kinds of weighted DP paths

Figure 5 illustrates some different weighted DP paths. If we use the path of Figure 5 (b), the path of slope 1/2 would not be often used. Thus, a spotted word location becomes a shorter part in a test pattern than the location detected by the path of Figure 5 (a). Conversely, if we use the path of Figure 5 (c) or (d), the path of slope 2 would be not often used. Thus, a spotted word location becomes a longer part.

Therefore, we use two kinds of DP paths such as Figures 5 (b) and (c) independently, we can obtain word spotting results with two different locations for each

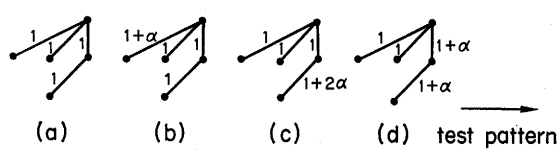

Fig. 5. Different local weighted DP paths.

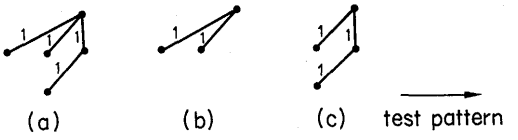

Fig. 6. Different local constrained DP paths.

ending frame in a test pattern.

(2) Two kinds of local constrained DP paths

Figure 6 illustrates some different local constrained DP paths. Figure 6 (a) is a basic DP path and it has a local constraint of slope 1/2 to 2. Figure 6 (b) is 1/2 to 1 and Figure 6 (c) is 1 to 2. If we use the DP path of slope 1/2 to 3/2, we can expect that a spotted word location becomes a longer part in a test pattern than the location detected by the basic DP path. This constraint is implemented easily by combining DP paths of Figures 6 (a) and (b). For example, we may use the DP path of Figure 6 (a) at odd frames in a test pattern and DP path of Figure 6 (b) at even frames.

Conversely, if we use the DP path of slope 2/3 to 2, a spotted word location becomes a shorter part than the location detected by the basic DP path. This constraint is also implemented easily by combining DP paths of Figures 6 (a) and (c). For example, we may use the DP path of Figure 6 (a) at the $3 \times n+1$st and $3 \times n+2$nd frames in a test pattern and DP path of Figure 6 (c) at the $3 \times$nth frame where $n=0, 1, \ldots, [I/3]$.

Thus if we use two kinds of DP paths with different local constraints independently, we can obtain word spotting results of two different locations for any ending frame, in a test pattern.

VII. Generation of Best Plural Candidates for Word Sequences

It is very useful to generate not only the best word sequence but the best C word sequences, that is, plural candidates. For example, even if the test sample is misrecognized as a different word sequence and if it is recognized within the top of best C word sequences, we need not utter the same word sequence again. We can choice the correct word sequence from C candidates by voice or typing. Another advantage is that the plural candidates have much information compared as only one candidate. For example, it is very powerful to represent a test sample by plural candidates of syllable recognition results (or syllable lattice) on the syllable-based word recognition [12].

The Level Building and $O(n)$ DP cannot generate the best alternative candidates. Therefore an approximate method is used [12] [13]. On the other hand, the Two Level DP and Augmented Continuous DP can generate the best C word sequences.

(1) Suboptimal plural candidates

We modify the equation (2) is subsection 2.3 as follows:

For $c=1, 2, \ldots, C$

$$\begin{aligned} m_n &= \underset{m}{\text{argmin}} \{D_{X-1}(m)+D^n(m+1:i)\} \\ \hat{n} &= \arg\ c\text{-th}\ \min_{n} \{D_{X-1}(m_n)+D^n(m_n+1:i)\} \\ D_X(i,c) &= D_{X-1}{}^{\hat{n}}(m_{\hat{n}})+D^{\hat{n}}(m_{\hat{n}}+1:i) \\ N_X(i,c) &= \hat{n} \\ B_X(i,c) &= m_{\hat{n}} \end{aligned} \tag{10}$$

where $D_X(i, 1)=D_X(i)$. The c-th $\min_n f(n)$ means the c-th minimum value of $f(n)$ for all n. In the above equation (10), only one best location is calculated for each reference, therefore the optimal best plural word sequences cannot be obtained.

This algorithm does not record only the best candidate for each ending frame of each x, but records the C best candidates. Thus, we may generate C different word sequences by backtracking. For example, we can generate the word sequence $W(c_1, c_2, \ldots, c_X)$, which the last word is the c_X -th candidate, the last -1 word is the c_{X-1}-th candidate, ..., and the first word is the c_1-th candidate, as follows:

1. $i=I$, $x=X$, $D=D_X(I, 1)$
2. $n=N_X(i, c_X)$ .....output
   $D=D+D_X(i, c_X)-D_X(i, 1)$
3. If $B_X(i, c_X)\neq 0$, $i=B_X(i, c_X)$, $x=x-1$. Go to 2.
   If $B_X(i, c_X)=0$, stop.

where D is the cumulative distance corresponding to the word sequence of $W(c_1, c_2, \ldots, c_X)$.

From these results, we can generate easily the best C word sequences by sorting cumulative distances given by D. It is considered simply as a tree-searching problem [13].

(2) Optimal plural candidates

Let $D_X(i, c)$ be the minimum cumulative distance for the c-th best word sequence when $a_1a_2\ldots a_i$ is matched with any concatenation of x reference patterns. We modify the equation (2) in subsection 2.3 as follows:

For $c=1, 2, \ldots, C$

$$\begin{aligned} D_X(i,c) &= c\text{-th}\ \min_{m,n,p} \{D_{X-1}(m,p)+D^n(m+1:i)\} \\ W_X(i,c) &= \hat{n} * W(\hat{m},\hat{p})\ ;\ \text{if } c_1\neq c_2, \text{ then } W_X(i,c_1)\neq W(i,c_2) \\ N_X(i,c) &= \hat{n} \\ B_X(i,c) &= \hat{m} \\ C_X(i,c) &= \hat{p} \end{aligned} \tag{11}$$

where $\hat{m}$, $\hat{n}$ and $\hat{p}$ satisfy the above equation, $W_X(i, c)$ means the c-th best word sequence, and the operation '$*$' means the concatenation of the word in the left hand operand and the word sequence in the right hand operand. The c-th best word sequence is given by $W_X(I, c)$. We can also generate $W_X(I, c)$ in the inverse order for the sequence as the following:

1. $i=I$, $x=X$, $p=c$
2. $n=N_X(i, p)$.....output
3. If $B_X(i, p)\neq 0$, $i=B_X(i, p)$, $p=C_X(i, p)$, $x=x-1$. Go to 2.
   If $B_X(i, p)=0$, stop.

The calculation of the modified equation is very simple and less computation for the Two Level DP or Augmented Continuous DP, because $D^n(m+1:i)$ is calculated firstly and then $D_{X-1}(m, p)+D^n(m+1: i)$ is calculated.

On the other hand, in the case of the Level Building or O(n) DP, the above equation (11) cannot be solved, therefore the equation may be modified in the following:

$$\text{For } c=1, 2, \ldots, C$$
$$m_{np}=\underset{m}{\operatorname{argmin}}\{D_{X-1}(m, p)+D^n(m+1: i)\}\ ;\ n=1, 2, \ldots, N;\ p=1, 2, \ldots, p.$$
$$D_X(i, c)=c\text{-th}\min_{m_{np}}\{D_{X-1}(m_{np}, p)+D^n(m_{np}+1: i)\} \tag{12}$$

Even if this modified equation is solved, the solution is still suboptimal and the amount of computation for cumulative distances becomes p times as comparison with the original algorithm.

VIII. Endpoint-Free Matching

Since the spectral patterns in word boundaries of an input pattern are uncertain or unstable because of word-juncture, we had better skip this region for matching with a reference pattern. In the O(n) DP matching algorithm, we obtained better recognition results by introducing the endpoint-free matching mechanism on an input pattern [7]. We also tried it for the Augmented Continuous DP matching algorithm. However, the formalism of the equation (1) shows that any input frame should be matched with a frame of a reference pattern. Therefore we propose a new endpoint-free matching algorithm which is matched an input frame in the skipped region with a pseudo pattern, that is, uses an estimated local distance or default distance instead of a matching distance. This idea was first introduced for the endpoint-free matching of the O(n) DP [7].

As described in chapter II, the original Augmented Continuous DP algorithm also uses an estimation mechanism. In this chapter, we propose a different estimation method from a view point of endpoint-free matching.

Original Algorithm

$$D(i)=\min_{n}\ [D(\bar{B}^n(i, J^n)-1)+\bar{D}^n(i, J^n)\times(i-\bar{B}^n(i, J^n)+1)/J^n],$$

New Algorithm

$$D(i)=\min_{n,r}\ [D(\bar{B}^n(i, J^n)+r)+\bar{D}^n(i, J^n)\times(i-\bar{B}^n(i, J^n))/J^n] \tag{13}$$
$$-\text{Dave}\times r+\text{Dpe}\times|r|$$

where Dave and Dpe denote an average local distance and penalty distance for skipping or overlapping, respectively. If $\text{Dave}=\bar{D}^n(i, J^n)/J^n$ and $\text{Dpe}=0$, the new algorithm equalls to the original algorithm.

IX. EXPERIMENTAL RESULTS ON CONNECTED DIGIT RECOGNITION

9.1. Speech Materials and Recognition Method

In order to test the Augmented Continuous DP algorithm, each of three male speakers spoke 10 isolated digits (two times) and 100 randomly generated strings of three digits. Ten or twenty isolated digits are used for making the reference patterns and 100 strings for test patterns. The speech signals are recorded low-passed to 4.4 KHz, and sampled at 10 KHz with 12 bit accuracy. These sampled data are preemphasized by using a $1-Z^{-1}$ filter and then analyzed by DFT within a Hamming window of 25.6 msec. Thus, fourteen cepstrum coefficients and the zeroth coefficient (power) are calculated at every 10 msec. The zeroth coefficient was used for the detection of silent parts. The other cepstrum coefficients were used for the calculation of local distances. The distance was defined by the city-block distance between two sets of coefficients.

We used the DP path of Figure 2 (c) for the Augmented Continuous DP and one or two reference patterns per digit. In the case of one reference pattern, we used only first 10 digit without averaging. All experiments were performed by a speaker-dependent mode.

Table 1 shows the rate of speech for these speakers. The rate of speaker HY is faster than others for utterances of test pattern.

9.2 Experimental Results

Table 2 summarizes the recognition results by the original Augmented Continuous DP algorithm [10]. The unknown length and known length denote the unknown case of number of words in a test pattern and the known case, respectively. The segmentation rate means the rate such that the number of words is correctly identified in a test pattern. The results were improved by the known length or by the double templates. These results were comparable with the results by traditional algorithms such as the Two Level DP and Level Building algorithms as described in the following subsection. Although the results were improved by using plural word spotting results for each ending frame in a test pattern ($K=2$), the rate was small.

Tables 3 and 4 summarize the recognition results by modified Augmented Continuous DP algorithms. Table 3 (a) shows that the string rate was improved to 85.7% from 83.7%. Specially, the consideration of silence in a spoken word improved the recognition performance. On the other hand, the consideration of silence at word boundaries did not improve the performance, because there was no

Table 1. Average duration per digit (ms)

Utterance \ speaker	SN	HY	TU
Reference pattern	333	344	350
Test pattern	329	279	324

Table 2. Recognition results for connected digits by original Augmented Continuous DP matching

speaker	recognition rate (%)		
	word	string	segment
(a) unknown length, K=1, one reference per digit			
SN	99.3	97.0	99.0
HY	93.0	70.0	88.0
TU	95.7	84.0	95.0
average	96.0	83.7	94.0
(b) known length, K=1, one reference per digit			
SN	99.3	98.0	(100.0)
HY	92.7	80.0	(100.0)
TU	95.7	87.0	(100.0)
average	96.3	88.3	(100.0)
(c) unknown length, K=2, one reference per digit			
SN	100.0	99.0	99.0
HY	93.7	73.0	88.0
TU	95.3	83.0	97.0
average	96.3	85.0	94.7
(d) unknown length, K=1, two references per digit			
SN	100.0	99.0	99.0
HY	95.3	74.0	85.0
TU	97.7	91.0	98.0
average	97.7	88.0	94.0
(e) known length, K=1, two references per digit			
SN	100.0	100.0	(100.0)
HY	94.3	85.0	(100.0)
TU	97.3	93.0	(100.0)
average	97.3	92.7	(100.0)

Table 3. Recognition results by modified Augmented Continuous DP matching (one reference per digit)

speaker	recognition result (%)		
	word	string	segment
(a) consideration of silence, unknown length, K=1			
SN	99.3	97.0	99.0
HY	90.3	74.0	89.0
TU	95.7	86.0	99.0
average	95.1	85.7	95.7

(b-1) normalization by word length and addition, unknown length, K=1

SN	99.0	96.0	99.0
HY	87.0	66.0	79.0
TU	94.7	84.0	94.0
average	93.6	82.0	90.0

(b-2) normalization by word length and addition, known length, K=1

SN	99.0	97.0	(100.0)
HY	94.0	84.0	(100.0)
TU	95.3	86.0	(100.0)
average	96.1	89.0	(100.0)

(c) normalization by word length and multiplication, known length, K=1

SN	99.0	97.0	(100.0)
HY	93.7	83.0	(100.0)
TU	95.3	86.0	(100.0)
average	96.0	88.7	(100.0)

(d-1) two kinds of DP paths (α=0.1), unknown length, K=2

SN	99.7	98.0	99.0
HY	91.0	73.0	88.0
TU	96.7	81.0	91.0
average	95.8	84.0	92.7

(d-2) two kinds of DP paths (α=0.2), unknown length, K=2

SN	99.3	97.0	99.0
HY	92.7	77.0	89.0
TU	97.0	80.0	87.0
average	96.3	84.7	91.7

(d-3) two kinds of DP paths (α=0.4), unknown length, K=2

SN	99.0	96.0	98.0
HY	94.3	78.0	88.0
TU	98.0	80.0	82.0
average	97.1	84.7	89.3

(e) two kinds of constrained DP paths, unknown length, K=2

SN	100.0	99.0	99.0
HY	91.3	73.0	87.0
TU	95.7	80.0	90.0
average	95.7	84.0	92.0

silence at word boundaries except for the silence at the front of 'kyu' (nine) in our test samples.

On experiments of scoring mechanisms for connected words, Tables 3 (b) and (c) show that the normalization of cumulative distance by the length of a reference pattern may improve the performance in the known length, but that it does not

improve the performance in the unknown length. From these results, we find the fact that we had better regard the matching result of longer reference as important than that of shorter reference in the unknown length. This has been implemented automatically in our algorithm or traditional algorithms, because the cumulative distance for each reference pattern depends on the length.

Tables 3 (d) and (e) show the recognition results by simple algorithms for detecting two word spotting candidates for each ending frame in a test pattern ($K=2$). These results are comparable with the results by the original algorithm which exactly detects two word spotting candidates (see Table 2 (c)). In the case of speaker HY, the string recognition rate was improved by using this method, for

Table 4. Recognition results with the best plural candidates (one reference per digit)
upper: only first candidate, middle: first and second candidates,
lower: first, second and third candidates

speaker	recognition rate (%)		
	word	string	segment
(a) suboptimal plural candidates, unknown length, K=1			
SN	99.3	97.0	99.0
	99.3	98.0	99.0
	99.3	98.0	99.0
HY	93.0	70.0	88.0
	94.7	84.0	99.0
	95.0	84.0	100.0
TU	95.7	84.0	95.0
	97.3	89.0	98.0
	97.3	90.0	99.0
average	96.0	83.7	94.0
	97.1	90.3	98.7
	97.2	90.7	99.3
(b) optimal plural candidates, unknown length, K=1			
SN	99.3	97.0	99.0
	99.7	99.0	100.0
	99.7	99.0	100.0
HY	93.0	70.0	88.0
	96.3	89.0	100.0
	98.0	94.0	100.0
TU	95.7	84.0	95.0
	97.3	90.0	99.0
	98.3	93.0	100.0
average	96.0	83.7	94.0
	97.8	92.7	99.7
	98.7	95.3	100.0

example, to 78% from 70% as shown in Table 3 (d-3). On the other hand, there were many insertion errors in the case of speaker TU by using this method. Especially, the 6 strings were only insertion errors of 'ni (two in English)' in Table 3 (c). They were caused by the shortest duration of the reference pattern (230 msec). It may be enough to detect only one word spotting candidate ($K=1$) in the recognition of short words such as digits. We think, however, it is necessary to detect two or more word spotting candidates for longer words to avoid mis-word spotting. In such a case, these simple algorithms will become useful.

Table 4 summarizes the results with the best plural word sequences. In the case of suboptimal strategy, the best two candidates ($C=2$) were recorded at each ending frame and the best 3 word sequences of any length were found in the case of sub-optimal strategy. On the other hand, the best 3 word sequences were found directly in the case of optimal strategy. Table 4 shows that the optimal strategy is better

Table 5. Recognition results by endpoint-free Augmented Continuous DP
first row: original Augmented Continuous DP
second row: endpoint-free Augmented DP (Dave=$\bar{D}^n$ (i, j)/J, Dpe=15)
third row: endpoint-free Augmented DP (Dave=100, Dpe=15)
forth row: original O(n) DP
fifth row: endpoint-free O(n) DP

speaker	recognition rate (%)		
	word	string	segment
SN	99.3	97.0	99.0
	99.7	98.0	99.0
	99.3	97.0	99.0
	99.7	95.0	95.0
	99.7	97.0	95.0
HY	93.0	70.0	88.0
	89.0	70.0	89.0
	91.0	75.0	91.0
	95.0	82.0	96.0
	95.3	84.0	96.0
TU	95.7	84.0	95.0
	95.3	85.0	98.0
	96.0	86.0	97.0
	96.3	76.0	85.0
	97.0	80.0	85.0
average	96.0	83.7	94.0
	94.9	84.7	94.7
	95.4	86.0	95.7
	97.0	84.3	92.0
	97.3	87.0	92.7

than the suboptimal strategy. Within the top 3 word sequences, the test word sequence and word were recognized correctly at the average rate of 95.3% and 98.7%, respectively. If an uttered word in a test word sequence is not spotted correctly by our word spotting algorithm, the test word sequence or word might not be recognized within the top of several choices. Therefore we can conclude that the word spotting algorithm for $K=1$ worked well in the experiment.

Table 5 shows the recognition results by the endpoint-free matching algorithms where the average local distance was about 100. We found from the table that the introduction of endpoint-free matching on both the Augmented Continuous DP and O(n) DP was very effective for connected word recognition.

9.3 Comparison with Other Algorithms

In the companion paper, we compared the original Augmented Continuous DP with the O(n) DP or the Two Level DP and found that both the average performances were almost the same [10]. However, the superiority or inferiority depended on the manner of utterances. In this subsection, we evaluate this Augmented Continuous DP from the different view point.

One of the most significant characteristics of this algorithm is the manner of concatenation of spotted words. As described in 3.3, this algorithm estimates $D^n(m+1:i)$ from a spotted result $\bar{D}^n(i, J^n)$ for $m=\bar{B}^n(i, J^n)-r_1$, $\bar{B}^n(i, J^n)-r_1+1$, ..., $\bar{B}^n(i, J^n)+r_2$. Therefore, this yields the estimation errors. Such errors may cause recognition errors. On the other hand, this estimation mechanism accepts the overlap or skip between words on the concatenation of words [10]. Such an overlap or skip may improve the recognition performance on connected word recognition [14].

We evaluate the manner of concatenation of spotted words in our algorithm by the following procedure:

1. We decide the best 3 word sequences for a test pattern, which do not include the correct word sequence (input sequence), by using the Augmented Continuous DP matching.
2. We make four new reference patterns by concatenating reference patterns of words which correspond to the best 3 word sequences and correct word sequence.
3. We match the test pattern with each new reference pattern by using DP path of Fig. 2 (b). This matching method is the same as that for isolated word recognition.
4. We decide the best word sequence as a recognition result from matching results of four reference patterns.

We should notice that the DP path in the above procedure is an asymmetric DP path with the basic axis on the reference. Therefore, we cannot apply traditional algorithms such as the O(n) DP, the Two Level DP and the Level Building DP to this procedure. We call this matching procedure as "Admissible DP matching"

Table 6. Comparison of Augmented Continuous DP, Admissible DP and O (n) DP
upper: recognition results by Augmented Continuous DP
middle: recognition results by Admissible DP
lower: recognition results by O(n) DP orTwo Level DP

speaker	recognition rate (%)		
	word	string	segment
SN	99.3	97.0	99.0
	99.7	99.0	100.0
	99.7	95.0	95.0
HY	93.0	70.0	88.0
	89.0	69.0	84.0
	95.0	82.0	96.0
TU	95.7	84.0	95.0
	93.7	81.0	99.0
	96.3	76.0	85.0
average	96.0	83.7	94.0
	94.1	83.0	94.3
	97.0	84.3	92.0

for convenience sake.

Table 6 summarizes the recognition results. For comparison, the results by the $O(n)$ DP are also shown in the table. The $O(n)$ DP yields the same results as the Two Level DP or the Level Building DP. The differences of performance between the Admissible DP and the $O(n)$ DP were caused only by the differences of DP paths, that is, Fig. 2 (b) and Fig. 1 (b), respectively.*

From the table, we find that the performance of the Augmented Continuous DP is equall to or higher than that of the Admissible DP. This fact suggests that the manner of concatenation of spotted words in our algorithm worked well.

X. CONCLUSION

In this paper, we modified the original Augmented Continuous DP algorithm from the view points of the following: 1) matching methods for a silence in speech, 2) scoring mechanisms for connected words, 3) simple word spotting algorithms for detecting two locations for each ending frame in a test pattern, 4) generation methods of the best plural candidates for word sequences and 5) endpoint-free matching mechanism. Then we applied these modified algorithms to connected spoken digit recognition and evaluated them.

We found from experiments as follows: First, the consideration of speech

* Although we used only the best 4 reference patterns for the Admissible DP, strictly speaking, we must use all reference patterns corresponding to all possible word sequences. If we use all possible patterns, the recognition rate by the Admissible DP might become worse than the above results. However, such an experiment was not practical.

silence improves the recognition performance. Secondly, we had better regard the matching result of longer reference as important than that of shorter reference in the unknown case of number of words in a test pattern. Thirdly, the recognition results by simple algorithms for detecting two word spotting candidates are comparable with the results by the original complex algorithm. Forthly, the optimal strategy for the generation of the best plural candidates of word sequences is better than the suboptimal strategy. The both Level Building algorithm and O(n) DP algorithm cannot apply the optimal strategy, but both the Augmented Continuous DP algorithm and Two Level DP algorithm can apply it. Fifthly, we propose an end-point-free matching algorithm and showed the effectiveness. Finally, we compared it with other traditional algorithms and found that our algorithm has almost the same performance as them.

The most significant advantage is less computation on the Augmented Continuous DP algorithm in comparison with the Two Level DP algorithm.

References

1) T. K. Vintsyuk, "Speech discrimination by dynamic programming," Cybernetics, Vol. 4, No. 1, pp. 52–57 (1968).
2) H. Sakoe and S. Chiba, "A dynamic programming approach to continuous speech recognition," in Proceedings of the 7-th ICA, (1971).
3) T. K. Vintsyuk, "Element-wise recognition of continuous speech composed of words from a specified dictionary," Cybenetics, No. 2, pp. 361–372 (1971).
4) H. Sakoe, "Two level DP matching—a dynamic programming based pattern matching algorithm for connected word recognition," IEEE Trans. Acoust. Speech Signal Process. Vol. ASSP-27, No. 6, pp. 588–595 (1979).
5) C. S. Myers and L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition, "IEEE Trans. Acoust. Speech Signal Process. Vol. ASSP-29, No. 2, pp. 284–297 (1981).
6) H. Sakoe and M. Watari, "Clockwise propagation DP-matching algorithm for connected word recognition (in Japanese)," Trans. Committee on Speech Research Acoust. Soci. Japan, S81-65 (1981).
7) S. Nakagawa, "Connected spoken word recognition algorithms by constant time delay DP, O(n) DP and augmented continuous DP matching," Information Science, Vol. 33, pp. 63–85 (1984).
8) J. S. Bridle et al., "An algorithm for connected word recognition," in Proceedings of the ICASSP (1982), firstly described in the proceedings of the NATO Advanced Study Institute (1981), J. P. Haton Ed. "Speech Analysis and Recognition," D. Reidel Publishing Company (1982).
9) H. Ney, "The use of a one-stage dynamic programming algorithm for connected word recognition," IEEE Trans. Acoust. Speech Signal Process. Vol. ASSP-32, No. 3, pp. 263–271 (1984).
10) S. Nakagawa, "A connected spoken word recognition algorithm by augmented continuous DP matching (in Japanese)," Trans. Inst. Elect. Comm. Engres. Japan, Vol. 67-D, No. 10, pp. 1242–1249 (1984).
11) S. Nakagawa and T. Nakamoto, "Speaker-independent large vocabulary word recognition based on syllable by syllable input (in Japanese)," Trans. Inst. Elect. Comm. Engrs. Japan, Vol. 65-D, No. 12, pp. 1558–1565 (1982).
12) S. Nakagawa, T. Umezaki and M. M. H. Jilan, "A continuous speech recognition method based on mono-syllable units and a spoken word recognition method from a classified syllable string (in Japanese)," Trans. Inst. Elect. Comm. Engrs. Japan, Vol. 68-D, No. 6, pp. 1296–1303 (1986).
13) C. S. Myers and L. R. Rabiner, "A comparative study of several dynamic time-warping algor-

ithms for connected-word recognition," Bell Sys. Tech. Jour. Vol. 60, No. 7, pp. 1389–1409 (1981).

14) K. Tajima, A. Tanaka and M. Komura, "Overlap and split of reference patterns for connected word recognition," Jour. Acoust. Soci. Japan (E), Vol. 7, No. 1, pp. 13–20 (1985).

(Aug. 31, 1986, received)