

A CONNECTED WORD RECOGNITION METHOD UTILIZING DTW AND A COARTICULATION MODEL

Shuichi ITAHASHI* , Yutaka HISAMATSU ** .

ABSTRACT

Coarticulation is one of the major factors that make speech recognition difficult. In conventional connected word recognition methods, a template for a connected word sequence is made by simply concatenating the templates of a single word. Therefore, some misrecognitions occur on account of this disregard for coarticulation. This study examines the influence of coarticulation in concatenating templates. The boundaries between words are smoothed so as to incorporate the influence of the tail of the preceding word on the head of the following word. The results of recognition experiments of 35 tokens of 4-digit-sequences show a recognition rate improvement of 5%, giving an 98% correct recognition rate.

INTRODUCTION

The method of dealing with coarticulation in connected word recognition is important. A template for a connected word sequence is made by simply concatenating the templates of a single word by conventional methods utilizing DTW. However, the template is not continuous at word boundaries, and is not always suitable for reference patterns. By introducing the effect of coarticulation at word boundaries, the template would be made more suitable by smoothed discontinuity.

In order to smooth a boundary, the previous word must be determined, and it must be kept unchanged until the DTW for a word is completed. Therefore we have adopted two methods by which the previous word can be determined, and have tried to bring a coarticulation model into these algorithms. We have adopted a critically damped 2nd order linear system for our coarticulation model. We consider only the effect from previous words to following words in this model.

In this study, we used LPC cepstra as feature parameters. The speech signal was sampled at 10kHz and quantized into 12 bits. The speech signal was analyzed every 10ms(=frame) after differentiation to emphasize high frequency regions. A recognition experiment was done for 3 male speakers. Reference patterns were prepared for each speaker.

COARTICULATION MODEL

We adopted a critically damped 2nd order linear system for the coarticulation model. The step responses of the model are shown in Fig.1 for several time constants. In a discrete system, the response of the model is calculated according to equation (1) (ref 1).

$$y(i+1) = (1-r)^2 x(i+1) + 2ry(i) - r^2 y(i-1), (1)$$

where $x(i)$ is the input and $y(i)$ the output;

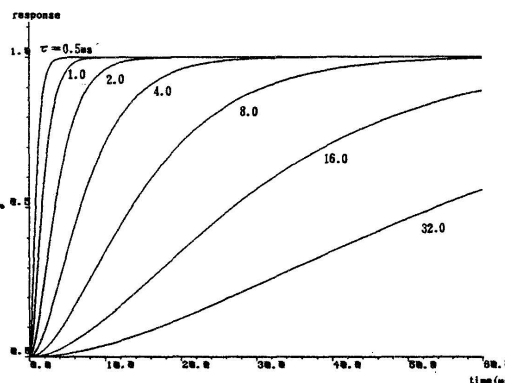


Fig.1 Step responses of the model

* Master's Program in Science and Technology, Graduate School, Univ. of Tsukuba

** Institute of Information Sciences and Electronics, University of Tsukuba, Japan

$r = \exp(-T/\tau)$; T is the sampling period (namely the frame interval, 10ms); and τ is the time constant. Using this response, the coarticulation model modifies the reference patterns to connect each element of the feature sets smoothly to the previous reference (see Fig.2).

The reference pattern $R_k(j)$ of word k is modified as follows.

$$R_k(1) = (1-r)^2 R_k(1) + 2r R_{ks}(J_{smp}) - r^2 R_{ks}(J_{smp}-1) \quad (2)$$

$$R_k(2) = (1-r)^2 R_k(2) + 2r R_k(1) - r^2 R_{ks}(J_{smp}) \quad (3)$$

$$R_k(j) = (1-r)^2 R_k(j) + 2r R_k(j-1) - r^2 R_k(j-2) : \text{for } j=3, \dots, 10 \quad (4)$$

where $R_{ks}(j)$ is the reference pattern of the previous word ks ; $J_{smp} = J(ks) - J_{bk}$; $J(ks)$ denotes the length of the word ks .

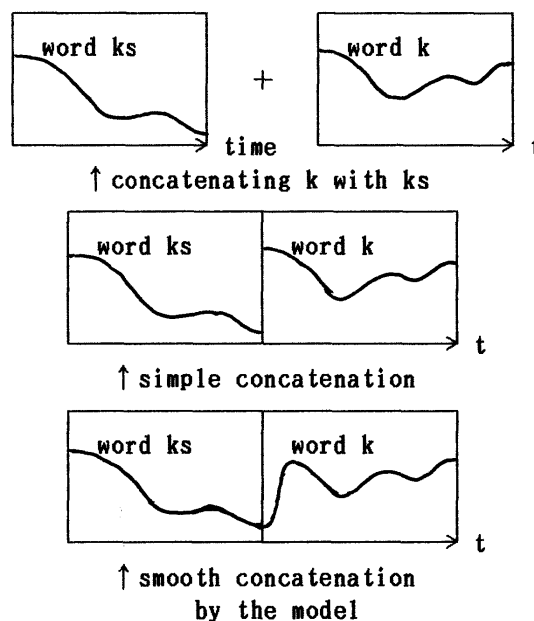


Fig.2 Smoothing the word boundary

In setting a large value to τ , the effect of the previous word ks becomes stronger, and at the limit of $\tau \rightarrow 0$, the effect disappears. We set τ at 10ms empirically.

J_{bk} specifies the start of the sampling position of the previous word. When sampling the feature parameters of the previous word ks , the last frame $R_{ks}(J(ks))$ is not always suitable for reference, because the decrease in the speech power in the last portion causes instability of the feature parameters. Therefore, J_{bk} is used for sampling the stable portion returning from the tail $J(ks)$ by J_{bk} . We fixed J_{bk} at 5.

In the formula (2) and (3), the feature parameters of the previous word ks are carried over, so as to connect the word k smoothly to the word ks . The modification is applied to the first 10 frames (0.1s) of the word k . Research on the modification range needs to be carried out in the future. The reference pattern k is modified when it is used. The modification has no influence on the original reference that is stored in a data file.

METHOD1: MODIFIED ONE-STAGE DYNAMIC PROGRAMMING ALGORITHM

We put the previous word loop into one-stage dynamic programming algorithm (ref 2), and accumulated distances were calculated for each previous word (see Fig.3).

When there is no previous word detected, the algorithm works as a one-stage DTW. Once previous words have been detected, the starting part of each reference pattern is modified to connect the previous word smoothly. This smoothing is done according to the coarticulation model explained above.

The algorithm is the following.

- 1) Initialize $D(0, j, k, ks) = 0$
and $D(i, j, k, ks) = \infty$, for all i, j, k, ks

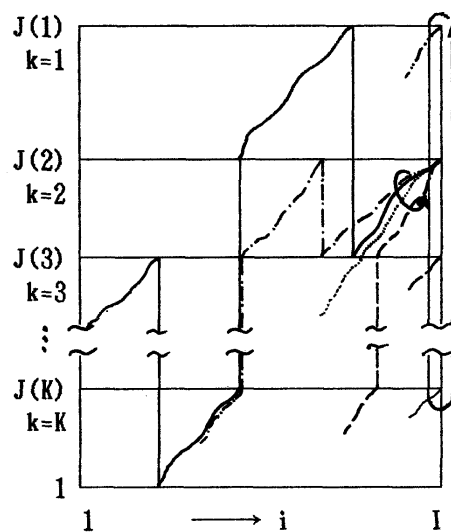


Fig.3 Modified one-stage DP

- 2) For $i=1, \dots, I$ do 3), 4), 5), 6), 7) and 8): test pattern frame loop
- 3) For $ks=1, \dots, K$ do 4), 5), 6) and 7): previous word loop
- 4) For $k=1, \dots, K$ do 5) and 6): word loop
- 5) For $j=1, \dots, J(k)$ do 6): reference pattern frame loop
- 6) $D(i, j, k, ks) = \text{minimum of the accumulated distance among allowed DP paths}$
- 7) $TS(i, ks) = \text{argmin}(k=1, \dots, K) D(i, J(k), k, ks)$,
 $FS(i, ks) = \text{starting point of } TS(i, ks)$
- 8) $T(i) = \text{argmin}(k=1, \dots, K) D(i, J(k), k, ks) : ks=1, \dots, K$,
 $F(i) = \text{starting point of } F(i)$
- 9) Backtrace optimal word sequence reversely from $T(I)$ using $T(i)$ and $F(i)$
 where $D(i, j, k, ks)$ is the accumulated distance at i -th frame of the test pattern, at j -th frame of the reference pattern k preceded by the word ks ; I is the test pattern length in frames; K is the number of words; $J(k)$ is the length of word k in frames; $TS(i, ks)$ is the optimal word following word ks at frame i ; $T(i)$ is the optimal word at frame i ; $\text{argmin}(x) f(x)$ denotes the argument x that minimizes $f(x)$.

We have adopted DP paths including a slope constraint. DP paths are different for the boundary and for the inside. At the boundary (i.e. $j=1$), two transitions are allowed: α_0, β_0 (see Fig.4). Both α_0 and β_0 refer to the accumulated distance of previous word ks , i.e.,

$$D(i, 1, k, ks) = \min \left\{ \begin{array}{l} D(i-1, J(ks), ks, ks1) \\ \quad + (CI+CJ) \cdot d(i, 1, k, ks), \quad : \alpha_0 \\ D(i-2, J(ks), ks, ks2) \\ \quad + (CI+CJ) \cdot d(i-1, 1, k, ks) \\ \quad + CJ \cdot d(i, 1, k, ks) \quad : \beta_0 \end{array} \right\} \quad (5)$$

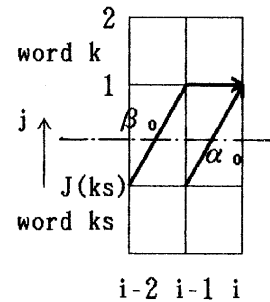


Fig.4 DTW paths between words

where $ks1=TS(FS(i-1, ks), ks)$; $ks2=TS(FS(i-2, ks), ks)$; CI is the weight for the test pattern; CJ is the weight for the reference patterns (we set CI at 1.0 and CJ at 0.0); $d(i, j, k, ks)$ is the local distance between frame i in the test pattern, and frame j in the reference pattern k preceded by the word ks . When ks is not detected, the accumulated distance terms (i.e. $D(*)$) are omitted. The transition β_0 is omitted when $i \leq 2$.

In the reference pattern k (i.e. $j \geq 2$), three transitions are allowed: α, β, γ (see Fig.5).

$$D(i, j, k, ks) = \min \left\{ \begin{array}{l} D(i-1, j-1, k, ks) \\ \quad + (CI+CJ) \cdot d(i, j, k, ks), \quad : \alpha \\ D(i-2, j-1, k, ks) \\ \quad + (CI+CJ) \cdot d(i-1, j, k, ks) \\ \quad + CI \cdot d(i, j, k, ks), \quad : \beta \\ D(i-1, j-2, k, ks) \\ \quad + (CI+CJ) \cdot d(i, j-1, k, ks) \\ \quad + CJ \cdot d(i, j, k, ks) \quad : \gamma \end{array} \right\} \quad (6)$$

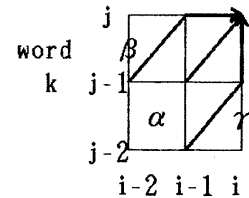


Fig.5 DTW paths within a word

When ks is not detected, the accumulated distance terms (i.e. $D(*)$) are omitted. The transition γ is omitted when $j=2$, and β is omitted when $i \leq 2$.

In the implementation of this algorithm, the range of the dimension i was reduced to 3 because the algorithm needs the accumulated distances of recent 2 frames of the test pattern (i.e. $i-1$ and $i-2$).

METHOD2: MODIFIED LEVEL-BUILDING DTW

We have developed a modified level-building DTW algorithm that searches for a quasi-optimal path between the test pattern and the optimal word sequence (ref 3). The DTW calculation of word k for level m is done from several starting points close to the optimal end point of each previous word (i.e. level $m-1$), and the optimal end point for level m is determined (see Fig.6). The end point of each word for level 0 is set at 0. For each previous word, the reference pattern is modified according to the coarticulation model.

RESULTS AND DISCUSSION

Table 1 shows the results of the tentative recognition experiment with METHOD1 (ref 5). The data used in the experiment contain 35 tokens of 4-digit-sequences that include all combinations of 2-digit-sequences. Each digit has only one reading, therefore we prepared 10 reference patterns for all digits and one more for 'silence'.

The table shows that the coarticulation model improves the recognition rate. It is noticeable that the accumulated distance is decreased by applying the coarticulation model. This indicates that coarticulation processing is a promising method.

A major problem with this method is the amount of calculation involved. It can be reduced by omitting those words whose accumulated distances are larger than a predetermined threshold.

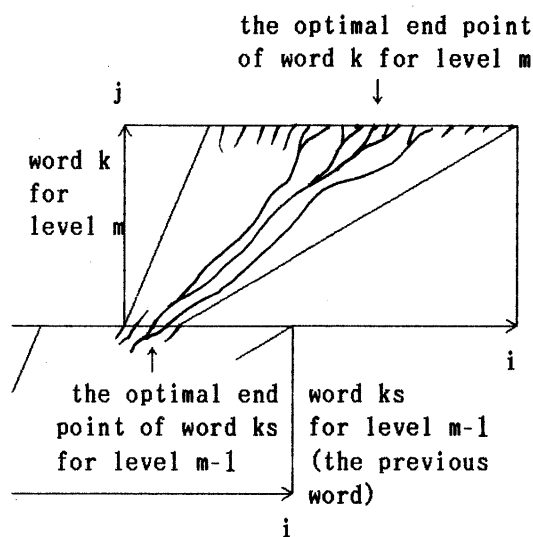


Fig.6 Modified Level-Building DTW

	AV.	S1	S2	S3
NOR	62.9	68.6	62.9	57.1
MOD	92.4	97.1	94.3	85.7
COA	98.1	100	100	94.3

Table 1. Correct rate (%)
(tentative recognition experiment)
NOR: normal one-stage DP
MOD: the weights of DP paths are modified
COA: with the coarticulation model

REFERENCES

1. ITAHASHI, Shuichi and Shoichi YOKOYAMA, "Description and Segmentation of Formant Trajectory with Second Order Linear System Model," (in Japanese), Bul. Electrotech Lab., Vol.40, No.6, 1976.
2. NEY, Hermann, "The Use of One-Stage Dynamic Programming Algorithm for Connected Word Recognition," IEEE Trans. ASSP-32, No.2, APRIL, 1984.
3. MYERS, C.S. and L.R.Rabiner, "A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition," IEEE Trans. ASSP-29, No.2, APRIL, 1981.
4. MYERS, C.S. and L.R.Rabiner, "A Comparative Study of Several Dynamic Time-Warping Algorithms for Connected-Word Recognition," The Bell System Technical Journal, Vol.60, No.7, September 1981.
5. HISAMATSU, Y and S.Itahashi, "A Method of Connected Word Recognition," (in Japanese), Tech. Group Speech, IECEJ, Paper SP86-25, July, 1986.