An Implementation of Recognition for Windows 95 Commander

Seong-kwon Lee* Soon-Hyob Kim*
*Department of Computer Engineering, Kwangwoon University, Seoul Korea

Abstract: This paper is a study on window95 commander recognition for implementation of real time voice system. The applied recognition algorithm is One-Stage /DPS with DMS(Dynamic MultiSection) model which generates reference pattern and recognizes 60 window commander and execute the commander. From the results of experiments, the optimal recognition considering a tradeoff of real time is obtained by applying DMS model with 20 section and 13 order PLP(Percepture Linear Prediction)coefficients for feature parameter. Window 95 Voice system is constructed by pentium 133MHz notebook. Voice data consists of 60 commander in frequently used. The references are made by 9 male speakers in the noisy laboratory environment.

The DMS Modeling Process

In the DMS Modeling, a word string splitted into some segments dynamically according to the sectional features, and adds to them the representative feature vector and time duration information for each segment. Therefore somewhat complicated procedure is required. It consists of two steps. The first step is dynamic segmentation, and the second is to extract representative feature vectors and segmental duration information.

Dynamic Segmentation Algorithm

The jth (1 ≤ j ≤ J, J is the number of each DMS segment) segment of DMS, M for each word has the segment information M(j). M(j) consists of feature vector \( m_j \) and segmental duration information \( P_j \). First, we segment the learning datum equally in the time axis, and gather feature vectors in its segment to get a centroid. This centroid is a representative feature vector of the segment. Supposing that all segments have same size, we divide the sum of the last frames in each segment by the total number of frames. By doing so, we can obtain segmental duration information. By applying DP matching algorithm to the learning data and the initial word model, we can shift the segment boundary dynamically. Finally, by getting total accumulated distance and assigning new segment in backtracking, we can get a new centroid in the new segment. This new centroid is a representative feature vector and model is updated. The ratio of the number of frames in each segment to total number of frames is recorded as a segmental duration information.

\[
D(i, j) = d_i(t_i, m_j) + \min \left\{ \begin{array}{l}
D(i-1, j), \quad (1 \leq i \leq I, 1 \leq j \leq J) \\
D(i-1, j-1) + P(j-1) \\
\end{array} \right. 
\]

\( P(j) \) is the difference distance between the ith frame of the learning data and the last frame of jth segment of word model.

\[
P(j) = W \times d_i(e(j), i) 
\]

\[
d_i(e(j), i) = | (e(j) \ast I - i) |
\]
Also, W is weighting value for the difference between a segmental duration information and an effecting element for a correct recognition rate. When training data T of 1 frame is divided into J section, a final frame of jth section is represented as e(j). Therefore, e(0)=0 and e(J)=1. In this paper, the distances between learning data T and word model M are computed by DP algorithm, and a segmental duration information is not used for recognition.

\[ t : \text{frame number of learning data } T \ (1 \leq i \leq l) \]

\[ j : \text{section number of word model } M \ (1 \leq j \leq J) \]

\[ d_i(t, m_j) : \text{local distance between feature vector } t_i \text{ in } i \text{th frame of learning data } T \text{ and feature vector } m_j \text{ in } j \text{section of model } M \]

\[ D(i, j) : \text{accumulated distances between } i \text{th frame of learning data and } j \text{th section of model} \]

**Recognition Experiment**

A speech data is sampled by 11.025 KHz, converted to 16bit data, pre-emphasized, covered by a Hamming window, and calculated by a 13's order autocorrelation. These calculate 13's order PLP cepstrum coefficient. The template is constructed by DMS. Speech DB consists of the 60 windows 95 commander. DMS model is experimented in speaker dependent and speaker independent mode. We split each word into 20 sections and make templates to reduce calculation time. In the word template, the shortest word marks about 40 frames and the longest word about 110 frames.

<table>
<thead>
<tr>
<th>Speaker</th>
<th>1st utterance Recognition Result</th>
<th>2nd utterance Recognition Result</th>
<th>3rd utterance Recognition Result</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>100.0 %</td>
<td>98.3 %</td>
<td>100.0 %</td>
<td>99.4 %</td>
</tr>
<tr>
<td>B</td>
<td>98.3 %</td>
<td>96.3 %</td>
<td>96.6 %</td>
<td>96.0 %</td>
</tr>
<tr>
<td>C</td>
<td>95.0 %</td>
<td>96.6 %</td>
<td>98.3 %</td>
<td>97.7 %</td>
</tr>
<tr>
<td>D</td>
<td>98.3 %</td>
<td>88.3 %</td>
<td>96.3 %</td>
<td>90.5 %</td>
</tr>
<tr>
<td>E</td>
<td>100.0 %</td>
<td>94.5 %</td>
<td>96.3 %</td>
<td>96.4 %</td>
</tr>
</tbody>
</table>

**REFERENCE**