

行動裝置上語音命令辨識系統之製作

Implementation of a Voice Command Recognition System for Mobile Devices

黃國勳

古鴻炎

Guo-Syun Huang and Hung-Yan Gu

國立台灣科技大學 資訊工程系

E-mail: guhy@mail.ntust.edu.tw

摘要

要在行動裝置上即時辨識語音命令，簡化及減少運算量是必需的，因此我們針對語音辨識的處理流程，研究了幾項把浮點運算轉換成定點運算的作法。此外，我們採取了隱藏式馬可夫模型(hidden Markov model)為基礎的辨識方法，但是把各個高斯混合(Gaussian mixture)的變異數參數去除，以提升維特比(Viterbi)搜尋的速度。模擬實驗之後，我們已把辨識程式移植到 PDA 上，實際測試顯示，該程式可以即時地進行七種語音命令的辨識，而辨識率約可達到 94%。

1. 前言

在行動裝置上，如果可以透過語音命令來控制、啟動一些動作(如自動撥號)，則可使行動裝置的操作變得方便許多。此外，大部分的家電是不會動的，且家電所需要執行的指令是有限的，如果要讓每樣家電都可以隨著語音命令起舞，實務上不可能在每樣家電旁邊擺著電腦，而最實際的作法就是在電器裡面裝設處理器(processor)，並且這樣的處理器的構造要越簡單越好，以節省費用，所以它只有少量的記憶體、小量的運算能力、消耗少量的電源。為符合上述硬體設備，小而快速的語音辨識軟體是必須的。

一般來說，一個語音辨識系統會包含二個重要的組件：特徵參數擷取、樣式(template)本比對。而樣本比對所使用的方法大體上有動態時間校準(dynamic time warping)、隱藏式馬可夫模型(hidden Markov model)、類神經網路(neural network)等三類[1, 2]。由於我們的研究目標是，製作小型的語音辨識系統，也就是花費少量的計算資源及少量的記憶體需求，因此我們選擇了 HMM 模型。這是因為 HMM 模型可以讓人彈性決定聲學模型的複雜度，當 HMM 模型的狀態數目，或各個狀態上的 Mixture 數目減少時，就可讓辨識處理所需的

記憶體減少，並且計算的複雜度也會跟著降低。另外，在本文的 HMM 模型中，我們也決定捨棄高斯機率函數中的變異數(variance)參數，以便減少計算量。至於動態時間校準之比對方法，雖然它與 HMM 模型一樣都要去作維特比搜尋，但是動態時間校準所用的聲學樣板缺乏彈性，在記憶體需求上遠大於 HMM 模型的。

在特徵參數方面，我們採用了一般常用的 MFCC (mel frequency cepstral coefficient)係數作為聲學特徵[1, 2]，對於 MFCC 係數的擷取，我們將研究基於定點(fixed point)運算的計算作法。雖然新式的小型裝置常常會附加浮點運算的功能，但是如果過分依賴浮點算器的話，語音辨識的應用，將會受到很大的侷限。除了考慮定點運算之外，我們還利用查表的方式來提高運算速度，把可能用到的數學函數，例如對數、三角函數等，都利用查表的方式或者是程式迴圈來製作，這樣的實作方式雖然會犧牲數值上的精確性，但是可以減少許多的計算量。

為了降低特徵參數擷取的計算量，我們採用了較低的取樣率(sampling rate) 8KHz，而不是一般常見的 16KHz 或是 22.05KHz。每個信號樣本，則以 16 位元(bits)來量化。至於特徵參數的維度，我們使用較少的 16 維，而不是常用的 26 或是 39 維。另外在端點偵測方面，我們採取以靜音(silence) HMM 模型來作端點偵測[1, 3]，而不是採用一般的短時能量或是越零率的分析[1, 2]。經由實驗的驗證，使用靜音 HMM 模型的辨識率表現，的確可以比使用短時能量及越零率的辨識率好一些。

2. 特徵參數擷取

觀察語音信號隨著時間變化的關係，可發現語音信號是時變的，所以必需把語音信號切割成一段一段的短暫片段，以便假設在一個短暫片段中，信號的特性是不變的[4]。這樣的短

暫片段稱之為音框(frame)，至於音框的長度，一般設定在 20~30ms 之間，不過在此我們採用的音框長度為 128 個樣本點，相當於 16ms，並且令相鄰的音框之間沒有重疊。

特徵參數的擷取以音框為單位來進行，其處理流程如圖 1 所示，分為兩大部分，一是前置處理，另一是特徵係數計算。前置處理可以再細分成三個部份”直流偏移移除”、”預強調”及”加窗處理”[1, 2]。特徵係數計算部分，也可細分成”傅利葉轉換”[4]及”梅爾倒頻譜”兩個部分[1, 2]。以下各子節中，將對各個細部處理方塊，說明我們的作法。

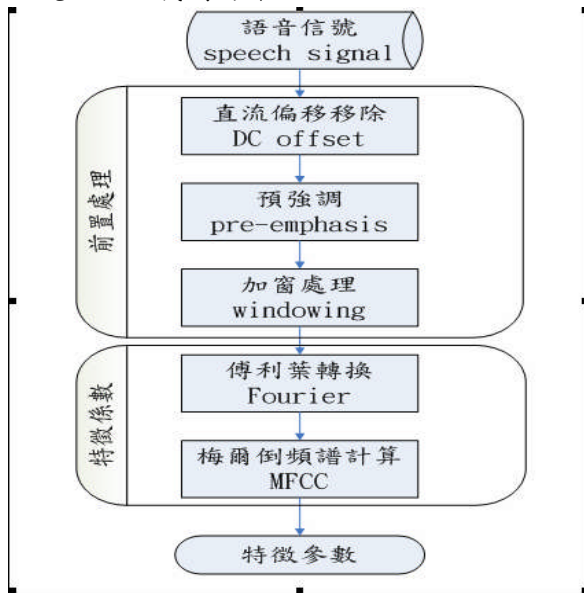


圖 1 特徵參數擷取之流程

2.1 直流偏移移除

麥克風錄音時，由於電子零件的關係，通常會發生直流偏移(DC offset)的現象，所以必需先扣除直流偏移的值。直流偏移值的原始算法是，加總音框內所有信號樣本的值再取平均。在本文中我們改成，對於一個音框求取它前面 16 個信號樣本值的平均值作為直流偏移，平均則以右移(Shift) 4 位元之運算來完成，公式為

$$dc = \left(\sum_{n=0}^{15} s(n) \right) \gg 4 \quad (1)$$

2.2 預強調

由於聲門(glottal)波形的影響，語音從嘴巴發出後高頻部分會被衰減，所以要用高通濾波器來加強高頻部分的信號，也就是預強調的處理，其原始處理公式為：

$$s_e(n) = s(n) - 0.95 \cdot s(n-1) \quad (2)$$

在本文裡為了進行定點計算，我們改成

$$s_e(n) = s(n) - \left[(243 \cdot s(n-1)) \gg 8 \right] \quad (3)$$

2.3 加窗處理

在時域中對語音信號作加窗的處理，則在頻域上會得到卷積(convolution)的平滑化效果。對於一個音框，當沒有特別作加窗的處理，實際上就是作矩形窗的加窗處理。除了矩形窗(rectangular window)之外，另一種常被使用的是漢明窗(Hamming window)，漢明窗的原始計算公式為：

$$W(k) = 0.54 - 0.46 \times \cos\left(\frac{2k\pi}{N-1}\right), 0 \leq k \leq 128-1 \quad (4)$$

在本文裡為了進行定點計算，我們改成

$$x(n) = (s_e(n) \times \text{HammingTable}[n]) \gg 8 \quad (5)$$

其中 HammingTable[n]儲存預先計算出的漢明窗第 n 個元素值，由於預先將漢明窗的數值放大了 256 倍，所以最後需再作右移 8 位元的運算。

2.4 傅利葉轉換

透過離散傅利葉轉換(discrete Fourier transform, DFT)[4]，可將時域信號轉成頻域的信號，即頻譜，轉換的公式為：

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk}, 0 \leq k < N \quad (6)$$

為了減少計算量，實作上都是使用快速傅利葉轉換(fast Fourier transform, FFT)演算法。令 $W_N = \exp(-j\frac{2\pi}{N})$ 和 $N = 2M$ ，可推得

$$X(k) = X_{\text{even}}(k) + X_{\text{odd}}(k) \cdot W_{2M}^k, k = 0, 1, 2, \dots, M-1 \quad (7)$$

其中 $X_{\text{even}}(k)$ 和 $X_{\text{odd}}(k)$ 的定義是：

$$X_{\text{even}}(k) = \sum_{n=0}^{M-1} x(2n)W_M^{nk} \quad (8)$$

$$X_{\text{odd}}(k) = \sum_{n=0}^{M-1} x(2n+1)W_M^{nk}$$

再以如此方式遞迴推導下去，就可得到快速傅利葉轉換的演算法[4, 5]。在 FFT 的計算過程，原先尤拉公式 $\exp(jw) = \cos(w) + j \cdot \sin(w)$ 所需的三角函數計算，本文將之修改為查表法，也就是

$$x(n) \cdot \cos(2\pi n) \Rightarrow (x(n) \cdot \text{cosTable}[2\pi n]) \gg 8$$

$$x(n) \cdot \sin(2\pi n) \Rightarrow (x(n) \cdot \sinTable[2\pi n]) \gg 8 \quad (9)$$

2.5 梅爾倒頻譜計算

本文採用梅爾倒頻譜係數(mel frequency cepstral coefficient, MFCC)，作為特徵參數。由於人類對於頻率的感知並不是線性的，所以傅利葉轉換所得的頻譜頻率，需依據梅爾頻率尺度來決定所屬的頻帶，線性頻率轉成梅爾頻率的公式為

$$Mel = 2595 \cdot \log\left(1 + \frac{f}{700}\right) \quad (10)$$

我們每間隔 100mel 放置一個三角形濾波器，來計算該濾波器所對應頻帶的信號強度 $Z_k, k=0, 1, \dots, 19$ 。接著，對各 Z_k 取對數(log) 而得到 Y_k ，關於對數函數的求值，我們將它製作成程式迴圈，計數右移(Shift) 多少次後會變成 1，所以實際上是取 2^n 中的 n 值。求得 Y_k 後，再經 cos 轉換公式

$$C_n = \sum_{j=0}^{19} Y_j \cdot \cos\left[n \cdot \left(j + \frac{1}{2}\right) \cdot \frac{\pi}{20}\right], \quad n=1, 2, \dots, 8 \quad (11)$$

就可求得倒頻譜係數 C_n ，在此 cos 轉換所需的 cos 函數值，我們也以如同公式(9)的查表法來取得。然後，本音框的 C_n 再減去前一音框的 C_n ，而得 ΔC_n 。

3. 隱藏式馬可夫模型之訓練方法

一個 HMM 模型需要三種機率參數來描述它，分別是 π, A, B ， $\pi = \{\pi_i\}$ 表示初始狀態機率(initial state probability)； $A = \{a_{ij}\}$ 表示狀態轉移機率(state transition probability)，轉移的方式只有兩種，停留在目前狀態或是轉移到下一個狀態； $B = \{b_j(O_t)\}$ 表示觀測機率(observation probability)，我們採用分割高斯混合(partitioned Gaussian mixtures model)來塑模 $b_j(O_t)$ ，由於是小型的語音辨識系統，為了減少計算量，我們捨棄了變異數。

在此 HMM 模型的訓練分成兩個階段，分別是初始模型(Initial Model)訓練，及分段 K 中心法(segmental K means, SKM)訓練[1]。

3.1 初始模型訓練

關於初始模型的訓練，訓練流程如圖 2 所

示，先將同一字彙的發音語料收集成一個群組，再依據人工標記語音的起點及結束點，將各發音的一序列音框作均等切分，而將音框平均分配到各個狀態，然後在各狀態上依分配到的特徵向量去訓練出一個高斯 mixture。

接著找出距離中心點最遠的一個特徵向量，作為另一個假設的中心位置，再將所有音框的特徵向量對這兩個中心位置作向量量化(vector quantization, VQ)，以求出新的中心點，重複上面的步驟直到中心點位置收斂為止，如此便可增加一個新的 mixture。這樣重覆作 mixture 擴增的處理，直到 mixture 數目達到所要求的為止，如此便可以訓練出 HMM 的初始模型。

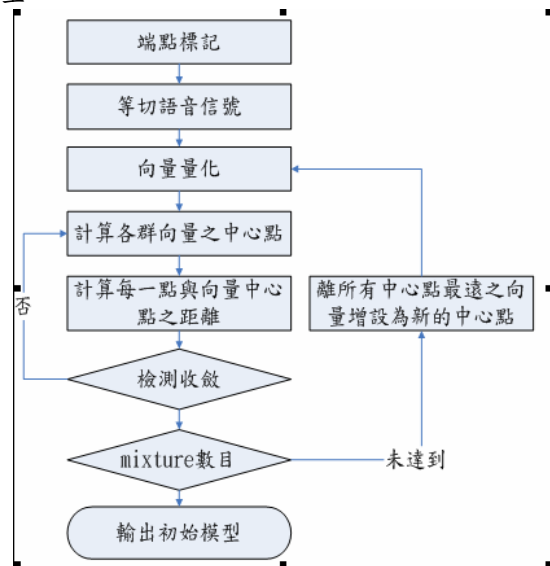


圖 2 HMM 初始模型之訓練流程

3.2 分段 K 中心訓練

初始 HMM 模型訓練完畢後，接著進行分段 K 中心的訓練，訓練流程如圖 3 所示。使用初始 HMM 模型來作維特比(Viterbi)搜尋之比對，在比對過程中可以得到各音框所停留的狀態及所屬的 mixture 的資訊，依據此資料作狀態及 mixture 的區分，就可收集屬於相同狀態、相同 mixture 的所有音框，來訓練出各個狀態上各個 mixture 的高斯分佈之參數，而建立出新的 HMM 模型。依此作法把新的 HMM 模型當成初始模型，再重複執行前述的步驟，一直到模型收斂為止。

3.3 模型參數設定之實驗

我們使用 PDA，收集 12 男 12 女的發音作為實驗用的資料，每位語者重複唸三遍如下的命令

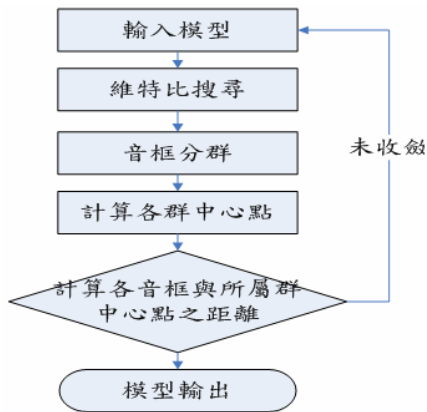


圖 3 分段 K 中心訓練的流程

，即”前進”、”後退”、”左轉”、”右轉”、”停止”、”準備”、”開始”。實驗目的是，了解最高的辨識率可達多少，以及 HMM 模型的狀態數和 mixture 數要設定為多少。實驗方式分為二種：(a)外部測試(outside test)、(b)內部測試(inside test)。這兩實驗所用的語音資料，都事先經過人工標記端點，所以語音檔案中不包含靜音信號。

(1) 外部測試: 輪流從 12 男 12 女的發音中，選取 10 男 10 女的發音作為模型訓練的語料，剩下的兩男兩女的發音就作為測試的語料，結果我們得到如表 1 所示的平均錯誤率數值，在設定狀態數為 10 且 mixture 數為 1 時，可得到最低的辨識錯誤率 6.1(即 93.9%之正確率)。此外，隨著狀態數的增加，辨識錯誤率會隨著降低，但是當狀態數超過 10 個以後，錯誤率便不再下降了。至於在 mixture 數方面，當狀態數低於 4 個的時候，mixture 數的增加具有改進辨識率的效果，但是當狀態數夠多以後，增加 mixture 數目對於辨識率的提升則沒有效果，反而會讓辨識率降低，這種現象的主要原因應是，音框的放置沒有重疊而使得訓練用的音框數目不夠多，mixture 數太大反而會讓某些 mixture 參數的訓練語料變得不足。

表 1 外部測試之狀態、mixture 組合的錯誤率

| % | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---------|------|------|------|------|------|------|------|------|------|------|------|------|
| | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. |
| 1 state | 69.7 | 53.2 | 46.2 | 41.3 | 38.7 | 38.3 | 37.5 | 36.5 | 37.3 | 36.5 | 35.6 | 34.5 |
| 2 state | 46.7 | 45.0 | 32.1 | 27.3 | 26.4 | 26.2 | 24.4 | 24.9 | 23.1 | 25.1 | 24.9 | 25.6 |
| 3 state | 32.8 | 31.4 | 21.0 | 17.3 | 16.9 | 19.2 | 19.0 | 20.1 | 21.6 | 20.5 | 21.6 | 22.3 |
| 4 state | 28.0 | 19.1 | 12.7 | 14.9 | 15.9 | 15.2 | 17.8 | 18.3 | 19.8 | 20.8 | 22.0 | 24.6 |
| 5 state | 17.2 | 11.6 | 12.3 | 14.7 | 16.1 | 18.7 | 20.4 | 22.2 | 23.7 | 24.1 | 26.7 | 24.6 |
| 6 state | 15.5 | 14.6 | 16.9 | 19.1 | 18.1 | 19.3 | 21.6 | 23.1 | 24.8 | 26.8 | 26.8 | 27.8 |
| 7 state | 11.8 | 16.5 | 13.1 | 16.9 | 20.4 | 21.6 | 23.4 | 27.2 | 28.3 | 29.9 | 31.0 | 30.2 |
| 8 state | 9.0 | 14.6 | 12.5 | 16.0 | 24.2 | 25.6 | 26.3 | 26.6 | 28.3 | 29.8 | 29.1 | 32.7 |
| 9 state | 7.3 | 12.5 | 12.0 | 19.7 | 25.3 | 29.4 | 29.3 | 28.7 | 30.1 | 29.1 | 30.5 | 31.7 |

| | | | | | | | | | | | | |
|---------|-----|------|------|------|------|------|------|------|------|------|------|------|
| 10 stat | 6.1 | 11.7 | 11.7 | 18.4 | 24.0 | 26.6 | 29.4 | 28.2 | 28.5 | 31.9 | 31.7 | 32.5 |
| 11 stat | 6.7 | 10.3 | 16.5 | 21.4 | 26.5 | 30.3 | 29.8 | 30.7 | 32.8 | 32.3 | 34.3 | 31.7 |
| 12 stat | 6.2 | 12.2 | 16.5 | 24.6 | 28.2 | 30.5 | 32.5 | 30.8 | 27.4 | 28.6 | 30.7 | 33.4 |

(2) 內部測試: 使用 12 男 12 女每人兩遍的發音作為訓練語料，另一遍的發音則作為測試語料，結果得到如表 2 所示的平均錯誤率數值，此實驗結果整體上與外部測試的結果相似，在設定狀態數為 10 且 mixture 數為 1 時，可得到最低的辨識錯誤率 3.6(即 96.4%之正確率)。此外，隨著狀態數的增加，辨識錯誤率會隨著降低。至於 Mixture 數增加時，卻只有在狀態數目少的時候，才有提高辨識率的效果。所以，狀態數的增加，對於辨識率的提升才會比較顯著的效果。

表 2 內部測試之狀態、mixture 組合的錯誤率

| % | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---------|------|------|------|------|------|------|------|------|------|------|------|------|
| | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. | mix. |
| 1 state | 69.0 | 50.0 | 45.2 | 43.4 | 39.9 | 39.9 | 44.0 | 44.0 | 42.9 | 35.1 | 35.1 | 39.9 |
| 2 state | 45.2 | 50.0 | 31.0 | 24.4 | 26.8 | 30.4 | 15.5 | 17.9 | 16.7 | 17.9 | 16.0 | 19.1 |
| 3 state | 29.8 | 22.0 | 14.9 | 14.3 | 17.9 | 16.1 | 16.1 | 12.5 | 16.1 | 14.9 | 18.5 | 14.3 |
| 4 state | 26.2 | 18.4 | 12.5 | 16.7 | 17.3 | 21.4 | 19.1 | 17.9 | 17.3 | 20.2 | 22.0 | 16.7 |
| 5 state | 17.9 | 10.7 | 14.3 | 23.8 | 11.9 | 23.2 | 17.9 | 28.6 | 22.0 | 15.5 | 26.8 | 22.0 |
| 6 state | 17.9 | 17.0 | 19.1 | 15.5 | 22.0 | 17.3 | 16.1 | 18.5 | 20.2 | 23.8 | 23.8 | 14.9 |
| 7 state | 10.1 | 11.9 | 10.7 | 11.9 | 25.6 | 21.4 | 32.7 | 22.6 | 32.1 | 19.0 | 31.6 | 20.8 |
| 8 state | 8.3 | 23.8 | 16.7 | 10.1 | 9.5 | 11.9 | 43.5 | 29.2 | 17.3 | 26.2 | 26.2 | 31.6 |
| 9 state | 5.6 | 12.5 | 19.6 | 28.6 | 10.7 | 31.0 | 16.1 | 32.1 | 26.2 | 20.2 | 26.8 | 32.7 |
| 10 stat | 3.6 | 8.9 | 23.2 | 14.3 | 4.2 | 33.3 | 23.8 | 27.4 | 20.2 | 14.2 | 36.3 | 29.2 |
| 11 stat | 5.9 | 26.8 | 5.4 | 14.9 | 15.7 | 10.1 | 24.4 | 23.8 | 9.5 | 14.9 | 23.2 | 19.1 |
| 12 stat | 6.0 | 15.5 | 25.6 | 24.4 | 39.9 | 22.0 | 13.1 | 22.0 | 22.0 | 12.5 | 28.0 | 35.1 |

4. PDA 語音命令辨識系統

4.1 系統發展

我們使用華碩 MyPal A730W Pocket PC 作為硬體配備，它內部使用的 CPU 是 Intel PXA270 520 MHz；作業系統是 Microsoft Windows Mobile 2003(MWM2003)第二版；記憶體包括 64MB flash ROM、128MB SDRAM；顯示介面是 VGA 640x480 的解析度；內建有可作錄音的麥克風。

關於軟體開發的配備，我們使用微軟公司的 eMbedded Visual C++ Service Pack 4, Pocket PC 2003 SDK, MWM2003 之 Developer Resources, MWM2003 之 Emulator Images for Pocket PC – WWE 等，來開發、製作我們系統的使用者介面。

語音特徵參數的擷取，主要是使用 Visual Studio 6.0 Service Pack5 來開發程式，程式寫作使用 C 語言。此部分的程式可共用於 Pocket PC

及一般 PC 上，因為裡面沒有使用到浮點或是特殊函式的運算，所以要移植到各種電腦平台都非常方便。

在 Pocket PC 上的錄音動作，我們以呼叫 eMbedded Visual C++ 所提供的多媒體控制函式來完成，函式的定義在 mmsystem.h 的檔案中可以查詢到。在多媒體的控制上，Pocket PC 的呼叫方式與一般 PC 上的呼叫方式相同，所以在 Pocket PC 上的辨識系統程式只要稍作修改，便可以直接在一般 PC 的視窗系統中使用。

4.2. 端點偵測

靜音 (silence) 是一種具有不同特性的信號，也可以使用 HMM 模型來描述它，當採取此種作法時，就是所謂的靜音模型之端點偵測。在此，我們將靜音建立成單個狀態而有多個 mixture 的 HMM 模型，不過由實際的實驗結果顯示，單個 mixture 之靜音模型還是具有較好的效能，亦即偵測出的端點與真實端點之間的時間誤差較小。

將靜音的 HMM 模型接在 N 個狀態之語音 HMM 模型的前後，便可以形成如圖 4 所示的複合式 HMM 模型，然後利用信號在複合 HMM 模型裡的狀態停留情況，來判斷語音信號的開始和結束，判斷規則為：若語音還沒發音，則目前的信號音框停留在靜音模型上的機率會比停在語音模型上的機率來得高；而當語音開始發音之後，則停留在語音模型上的機率會變得較高。應用此種規則，再去找出目前音框的最大機率之駐留狀態，就可以判斷語音信號是否開始發音了，還是結束發音了。

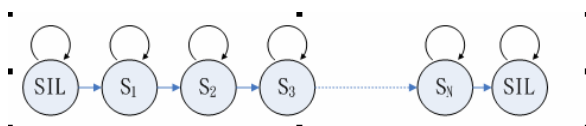


圖 4 語音 HMM 模型與靜音模型之串接方式

4.3 操作界面

我們辨識系統的操作介面如圖 5 所示，採取的是“Press and Talk”的方式，雖然有作端點偵測，但是為了避免讓 PDA 一直處於辨識狀態而消耗電能，所以必須先手動開啟語音辨識系統，再以筆去觸壓螢幕上的 Recog 按鈕來開始辨識。在圖 5 操作介面的其它部分，左上角長方塊用以選擇聲學模型，左側是語音命令列表，右上角是啟動語音辨識之按鈕，右側為語音辨識結果的列表。

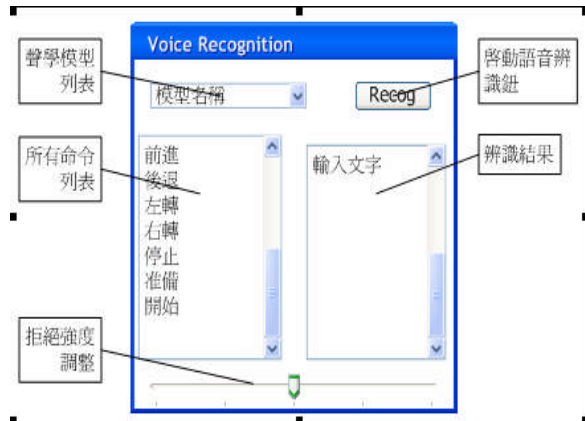


圖 5 PDA 語音命令辨識之操作介面

4.4 線上辨識實驗

在線上辨識實驗裡，我們找了 4 男 4 女實際使用 PDA 來作線上的語音命令辨識之測試，每個人對{“前進”、“後退”、“左轉”、“右轉”、“停止”、“準備”、“開始”}的各個命令分別發音三次。這些語者之中，2 男 2 女參加過 HMM 模型訓練語料的錄音，所以屬於內部測試；另外 2 男 2 女則未提供過訓練語料，所以屬於外部測試。

內部測試裡，2 男 2 女總共作了 84 次的發音，其中 2 次發生辨識錯誤，所以錯誤率是 2.4%。外部測試裡，一樣作了 84 次的發音，總共發生了 8 次的辨識錯誤，所以錯誤率是 5.9%。

在辨識速度方面，由於我們系統的製作，採用了靜音 HMM 模型而不必另外作端點偵測，並且錄音和特徵參數求取、HMM 模型機率計算是以多執行緒方式同時進行，所以當語者發完音時，HMM 模型的機率值也已經計算出來，也就是說語音命令的辨識是幾乎沒有延遲地即時完成。

5. 結論

為了在行動式或嵌入式裝置上製作語音命令辨識的功能，我們研究把特徵參數擷取的計算，由一般的浮點式計算改造成定點式計算；此外，還將牽涉到的數學函數的計算，改造成查表或是程式迴圈的作法，以提升處理的速度。

我們研究將 HMM 模型加以簡化，捨棄了變異數參數，讓語音特徵向量序列在跟模型作比對的時候，可以降低計算量，以加快處理的速度。此外，我們也對 HMM 模型的訓練作簡

化，只留下初始模型及分段 K-mean 中心法的訓練，這樣地簡化模型的訓練，是準備以後可以在行動式裝置上，直接作 HMM 聲學模型的訓練。

雖然我們使用了簡化的定點式計算，並且 HMM 模型也作了簡化，不過語音命令的辨識率仍可達到約 94%，這表示我們的方法在行動裝置上是可行的。

參考文獻

- [1] Rabiner, L. and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice-Hall, 1993.
- [2] O'Shaughnessy, D., *Speech Communication:*

Human and Machine, 2nd ed., IEEE Press, 2000.

- [3] 楊永泰，*隱藏式馬可夫模型應用於中文語音辨識之研究*，中原大學資訊工程學系，碩士論文，1999。
- [4] Oppenheim, A. V., R. W. Schaffer, and J. R. Buck, *Discrete-Time Signal Processing*, 2nd ed., Prentice-Hall, 1999.
- [5] 楊茗惠，*用隱藏式馬可夫方法於頻域特徵之國語數字辨識*，中興大學應用數學系，碩士論文，2002。