DOI:10.16526/j.cnki.11-4762/tp.2004.05.022

文章编号:1671-4598(2004)05-0470-03 中图分类号:TP31 文献标识码:B

MATLAB 环境下的基于 HMM 模型的语音识别系统

郭圣权,连晓峰
(华北工学院 自动控制系,山西 太原 030051)

摘要:在MATLAB环境下利用语音工具箱Voice Box实现基于连续概率密度隐含马尔科夫模型的汉语语音识别系统。在实时录音的情况下,利用该语音识别系统,不同的人对20条2~8个字的语音命令进行识别,准确率可达到95%,识别时间1.5~3 s,实现了小词汇量连续语音的非特定人的实时识别。

关键词:语音识别;MATLAB;连续概率密度隐含马尔科夫模型(CDHMM)

Speech Recognition System Based on HMM in MATLAB Environment

Guo Shengquan, Lian Xiaofeng
(Dept. of Automatic Control, North China Institute of Technology, Taiyuan 030051, China)

Abstract: Using Voice-Box a mandarin speech recognition system is realized based on CDHMM in MATLAB environment. Under the circumstance of the real-time recording, different people to say the connected words which include 2~8 words in 20 speech utterances, the accuracy can reach 95%, and the recognition time is between 1.5~3 seconds in this system.

Key words: speech recognition; MATLAB; Continuo's Density Hidden Markov Model

0 引言

目前,用于语音识别的方法有基于动态时轴归整(DTW)技术的模板匹配法、基于概率统计的HMM法和基于人工神经网络(ANN)的最优搜索法。

1 系统过程设计

1.1 实时录音和端点检测控件[1]

ActiveX控件是Microsoft公司制订的一种软件接口标准,在MATLAB中,通常是将ActiveX控件嵌入到figure窗口中,以GUI程序的方式使用。通过ActiveX录音控件可实现与MATLAB主程序的双向交互。当录音控件录音完成并进行端点检测后,向主程序发送一个事件,通知主程序读取语音数据及短时参数信息等.端点检测算法主要采用短时平均能量和短时平均过零率作为判决的主要特征。

$e(i)=\sum_{n=1}^{N}\log x_i^2(n)$ 计算第 i 帧的短时能量;

$zcr(i)=\sum_{n=1}^{N-1}|x_i(n)-x_i(n+1)|$ 计算第 i 帧的过零率

本系统采用8 kHz采样频率、16 bit、单声道的PCM录音格式,帧长 N 为30 ms,帧移 M 为10 ms,$x(n)$ 为输入的语音信号,x 为第 i 帧的第 n 个样本。系统界面如图1所示。

1.2 语音信号特征参数提取[1-2]

语音信号特征参数的提取是语音识别的一个重要环节。目前常用的方法是基于人的发音器官建立声道模型和基于听觉器官建立听觉模型。基于听觉模型得到的MEL倒谱系数(Mel-Frequency Cepstrum Coefficients,MFCC)比基于声道模型得到的LPC倒谱系数更符合人耳的听觉特性,在有信道噪声和频谱失真的情况下,能产生更高的识别精度。

MFCC建立在Fourier频谱分析基础上,首先利用人耳的感知特性,在语音的频谱范围内设置若干个带通滤波器,每个滤波器具有三角形或正弦形滤波特性,然

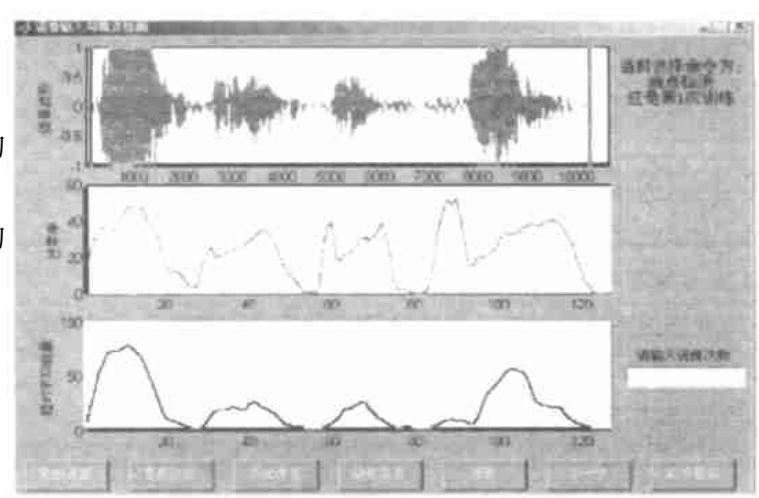

图1 语音输入与端点检测界面

收稿日期:2003-08-10
基金项目:山西省自然科学基金资助项目
作者简介:郭圣权(1939-),男,山西省定襄县人,教授,主要从事导航、制导及语音识别在导航定位系统中的应用。

后在特征矢量中纳入能量信息，计算相应滤波器组的信号能量，再通过离散余弦变换（DCT）计算其对应的倒谱系数。

语音信号的 MFCC 特征参数主要反映语音的静态特征，语音信号的动态特征可以通过这些静态特征的差分谱来描述，这些动态信息和静态信息形成互补，能很大程度上提高系统的识别性能。因此本系统利用 12 阶 MFCC 参数及其一阶差分作为语音信号的特征参数，系统界面如图（2）所示。

1.3　非特定人的语音识别算法——HMM 方法[2-3]

隐含马尔科夫模型（Hidden Markov Model）是 20 世纪 80 年代在语音识别领域的重大发展，一方面用隐含的状态对应声学层各相对稳定的发音单位，并通过状态转移和状态驻留描述发音的变化；另一方面引入了概率统计模型，用概率密度函数计算语音特征参数对 HMM 模型的输出概率，通过搜索最佳状态序列，以最大后验概率为准则找到识别结果。它较为完整地表达了语音的声学模型，并采用统计的训练方法将底层的声学模型和上层的语言模型结合到统一的语音识别搜索算法中。目前，基于 HMM 模型的语音识别算法是主流研究方向。

语音识别中使用 HMM 通常是用从左向右单向、带自环、带跨越的拓扑结构来对识别基元建模，一个音素就是一个三至五状态的 HMM，一个词就是构成词的多个音素的 HMM 串行起来构成的 HMM，而连续语音识别的整个模型就是词和静音组合起来的 HMM。本系统中采用连续概率密度隐含马尔科夫模型（Continuo’ s Density Hidden Markov Model，CDHMM）对每条命令进行训练。

1.3.1　连续概率密度隐马尔可夫模型

CDHMM 模型的每个状态观测概率密度函数由 n 个连续高斯概率密度函数（pdf）的线性组合来描述，每个 pdf 都有各自的均值矢量和协方差矩阵。因此 HMM 模型 $\lambda=(A, B, \pi)$ 中的输出概率密度函数：

$$B=\{b_j(o)\},\ b_j(o)=\sum_{l=1}^{M} c_{jk} N(o, \mu_{jk}, U_{jk}) 1\leqslant j\leqslant N$$

其中，o 表示给定的观测矢量，M 为每个状态包含的高斯元个数，c_{jk} 为第 j 状态下第 k 个混和高斯函数的权，N 代表正态高斯概率密度函数，μ_{jk} 和 U_{jk} 分别为第 j 状态下第 k 个混和高斯元的均值矢量和协方差矩阵。这里，权系数 c_{jk} 满足以下条件：

$$\sum_{k=1}^{M} c_{jk}=1 \quad 1\leqslant j\leqslant N \quad 1\leqslant k\leqslant M \quad c_{jk}\geqslant 0$$

1.3.2　HMM 模型的三项基本问题的解决

（1）输出概率的计算问题：给定观察序列和 HMM 模型，通过前向概率和后向概率计算观察序列对 HMM 模型的输出概率。

HMM 的前向概率为：

$$\alpha_t(i)=P(o_1 o_2 \cdots o_t, q_t=i \mid \lambda)$$

表示给定 HMM 模型参数，部分观测序列 $\{o_1 o_2 \cdots o_t\}$ 在 t 时刻处于状态 i 的概率。

它的递推计算公式如下：

a．初始化

$$\alpha_1(i)=\pi_i b_i(o_1) \quad 1\leqslant i\leqslant N$$

b．迭代计算

$$\alpha_{t+1}(j)=\left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1})$$

$$1\leqslant t\leqslant T-1 \qquad 1\leqslant j\leqslant N$$

c．终止计算

$$P(o \mid \lambda)=\sum_{i=1}^{N} \alpha_T(i)$$

相应地，HMM 的后向概率为：

$$\beta_t(i)=P(o_{t+1} o_{t+2} \cdots o_T, q_t=i \mid \lambda)$$

表示给定 HMM 模型参数，观测序列在 t 时刻处于状态 i，系统输出部分观测序列 $\{o_{t+1} o_{t+2} \cdots o_T\}$ 的概率。

它的递推计算公式如下：

a．初始化

$$\beta_T(i)=1 \qquad 1\leqslant i\leqslant N$$

b．迭代计算

$$\beta_t(i)=\sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)$$

$$1\leqslant t\leqslant T-1 \qquad 1\leqslant i\leqslant N$$

由此，可以根据前向概率和后向概率得到整个观测序列对 HMM 模型的输出概率

$$P(o \mid \lambda)=\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)=$$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j) \quad 1\leqslant t\leqslant T-1$$

（2）状态序列解码问题：给定观察序列

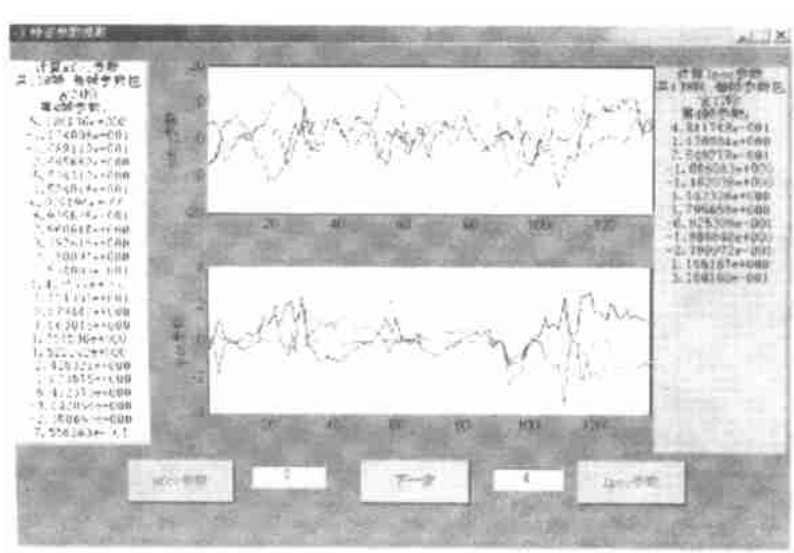

图 2　MFCC 参数与 LPCC 参数比较界面图

和 HMM 模型，通过 Viterbi 识别算法确定一个最优的状态转移序列。viterbi 算法是广泛用于通信领域的动态规划算法，它不仅可以找到一条”最优”的状态转移路径，还可以得到该路径所对应的输出概率。为减少计算量，通常采用对数形式的 Vterbi 算法。其算法如下：

a. 预处理

$$\tilde{\pi}_i = \log(\pi_i)$$

$$b_i(o_t) = \log[b_i(o_t)]$$

$$\tilde{a}_{ij} = \log(a_{ij})$$

b. 初始化

$$\tilde{\delta}_1(i) = \log[\delta_1(i)] = \tilde{\pi}_i + b_i(o_1)$$

$$\Psi_1(i) = 0 \qquad 1 \leqslant i \leqslant N$$

c. 递推计算

$$\tilde{\delta}_t(j) = \log[\delta_t(j)] = \max_{1 \leqslant i \leqslant N}[\tilde{\delta}_{t-1}(j) + \tilde{a}_{ij}] + b_j(o_t)$$

$$2 \leqslant t \leqslant T$$

$$\Psi_t(j) = \arg\max_{1 \leqslant i \leqslant N}[\tilde{\delta}_{t-1}(j) + \tilde{a}_{ij}] \qquad 1 \leqslant j \leqslant N$$

d. 终止计算

$$P^* = \max_{1 \leqslant i \leqslant N}[\tilde{\delta}_T(i)]$$

$$q_T^* = \arg\max_{1 \leqslant i \leqslant N}[\tilde{\delta}_T(i)]$$

e. 回溯最佳路径

$$q_t^* = \Psi_{t+1}(q_{t+1}^*) \qquad 1 \leqslant t \leqslant T-1$$

这里，$\delta_t(i)$ 为 t 时刻第 i 状态的累积输出概率，Ψ_t 为 t 时刻第 i 状态的前续状态号，q_t^* 为最优状态序列中 t 时刻所处的状态，P^* 为最终的输出概率。

（3）模型参数的估计问题：利用 Baum－Welch 算法训练模型参数，使观察序列对 HMM 模型的输出概率最大。Baum–Welch 算法主要利用极大似然（ML）准则对初始化的 HMM 模型参数进行重估，即对同一条命令用不同的人说多次，分别计算出各自的特征参数序列，然后用其重估模型参数。在本系统中，用户可以在训练流程中对每条命令训练 P 次，则记录训练次数 P，由此生成的相应的 P 个观测序列可构成集合：

$$O = [O^{(1)}, O^{(2)}, \cdots, O^{(p)}, \cdots, O^{(P)}]$$

其中 $O^{(p)}$ 为第 p 个观测序列，时间长度为 T_p：

$$O^p = [O_1^{(p)}, O_2^{(p)}, \cdots, O_{T_p}^{(p)}]$$

定义过渡概率 $\xi_t(i, j)$ 为观测序列在 t 时刻处于状态 i，$t+1$ 时刻处于状态 j 的概率

$$\xi_t(i,j) = P(q_t = i, q_{t+1} = j \mid o, \lambda) = \frac{P(q_t = i, q_{t+1} = j, o \mid \lambda)}{P(o \mid \lambda)} = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(o \mid \lambda)}$$

定义混和输出概率 $\gamma_t(j, k)$ 为某个观测序列在 t 时刻处于状态 j 的对于第 k 个混和高斯元的输出概率

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}N(o_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^{M}c_{jm}N(o_t, \mu_{jm}, U_{jm})}\right]$$

由此可以得到 HMM 模型参数的重估公式：

$$\bar{a}_{ij} = \frac{\sum_{p=1}^{P}\sum_{t=1}^{T_p-1}\xi_t^{(p)}(i,j)}{\sum_{p=1}^{P}\sum_{t=1}^{T_p-1}\sum_{n=1}^{N}\xi_t^{(p)}(i,n)}$$

$$\bar{c}_{jk} = \frac{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma_t^{(p)}(j,k)}{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\sum_{m=1}^{M}\gamma_t^{(p)}(j,m)}$$

$$\mu_{jk} = \frac{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma_t^{(p)}(j,k)o_t^{(p)}}{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma_t^{(p)}(j,k)}$$

$$U_{jk} = \frac{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma_t^{(p)}(j,k)(o_t^{(p)} - \mu_{jk})(o_t^{(p)} - \mu_{jk})'}{\sum_{p=1}^{P}\sum_{t=1}^{T_p}\gamma_t^{(p)}(j,k)}$$

由于本系统采用从左至右、无跳转、单向结构的 HMM 模型，初始概率恒等于

$$\begin{cases} \pi_1 = 1 \\ \pi_i = 0 \qquad 2 \leqslant i \leqslant N \end{cases}$$

因此不需对它进行重估。

2 识别实验

本系统可接受最多 9 s 的含命令语音，对 8 kHz 采样频率、16 bits、单声道的 PCM 语音进行每帧 30 ms 的短时分析，采用 24 阶 MFCC 及一阶差分参数作为语音特征参数。创建包含 20 条 2～5 个词组成的语音命令数据库，对每条命令分别建立 4 个状态、每个状态含 3 个高斯概率密度函数的 CDHMM 模型。

系统分训练和识别两个流程。训练过程对 HMM 模型参数进行最多 20 次的迭代计算，直到收敛。实验表明，对不同命令的训练时间在 10～50 s。识别过程将测试语音的 HMM 模型与训练库中的各个模型应用 Viterbi 算法进行搜索，找到输出概率最大的训练模型。实验表明，识别时间在 1.5～3 s，识别准确率为 95％左右。

3 结束语

上述是一个简单的语音识别系统，可用于小词汇量的非特定人连续汉语语音的实时识别，该系统计算简单，识别时间短。但对于一个完善的语音识别系统，一方面需要改造成其它高级编译语言的程序，以提高执行速度；另一方面还需考虑 HMM 模型中参数共享问题以提高系统性能以及改善识别算法，提高系统的自适应能力。

（下转第 475 页）

1.4　软件设计

在适配卡的软件设计中，采用模块化的设计思想，图 3 为所设计的多个程序模块。

其中 TIMER0 定时中断程序用于适配卡的工作状态指示以及对 CAN 总线上节点进行定时巡检。

状态检测程序用于适配卡运行期间的自检程序，方法为在双口 RAM 以及数据缓冲器的特定单元设置状态标志，运行期间不断测试，以保证系统的可靠性运行。另外，80C592 芯片本身的 T3 定时器是专为防止单片机“死机”而设置的“看门狗”定时器，如果单片机因噪声或射频干扰而进入一种错误运行状态时，经过一段特定的时间后，T3 会自动发出一个系统复位信号，使单片机脱离错误状态、从而避免“死机”，80C592 本身具有看门狗使能端口/EW，设计时/EW 接地，“看门狗”功能有效且不能被软件取消；在状态检测程序中，需要给 T3 定时器装入看门狗时间间隔，相当于看门狗计数器清零。采用状态检测程序是保证适配卡正常工作的可靠性设计措施。

紧急信息处理程序用于适配卡处理报警等紧急数据，根据适配卡参数配置存储器中的设置，决定立即上传给 PC 或暂存，并决定采取何种措施。

2　应用程序及设备驱动程序的设计

2.1　应用程序设计

采用 Microsoft Visual C＋＋ 6.0 开发上层的应用软件，其中运用了面向对象的编程设计方法和多线程技术。PC 机应用程序中的类的划分如图 4 所示。程序中主要有 5 个类：

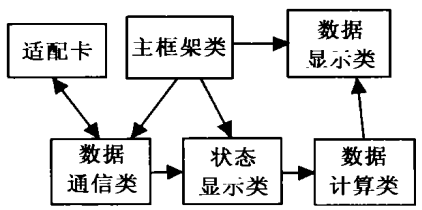

图 4　上层应用程序类模块划分图

在程序中，主框架类中定义了一些各类使用的公共数据和可视化程序框架的初始化工作。数据通讯类中定义了通讯的格式和对适配卡的通讯的各种操作。通讯状态显示类中封装了对通讯得到的数据进行处理的操作和通讯状态的显示。数据曲线显示类中封装了数据曲线动态或静态显示的各种操作等。数值计算类完成数据处理中各种数值计算。

2.2　设备驱动程序设计

这里采用 Numega 公司的 Driverstudio 软件开发了针对适配卡的虚拟设备驱动程序。

对于硬件中断编程，Driverstudio 提供了类 VhardwareInt 来实现某个 IRQ 端口的虚拟化，并处理该 IRQ 端口上的硬件中断。在设备驱动程序设计中主要采用了 VhardwareInt 类完成对中断的虚拟化。

2.3　应用程序与设备驱动程序的接口

设备驱动程序相当于一个硬件设备被虚拟化了，这个硬件设备可以作为一个文件系统而被系统调用。通过调用 Win32 函数 CreateFile 来打开或者创建“文件”，在应用程序在加载设备驱动程序之后，调用 DeviceIoControl 就可以与设备交换数据。

与此相应，在编写 VxD 时，程序需要提供一个相应的回调函数来响应它所发出的 DeviceIoControl 消息，并再去回调函数中响应相应的控制码。若要驱动程序支持 DeviceIoControl 的调用，必须为其发送的消息 W32_DEVICEIOCONTROL 建立相应的消息处理函数，在实际的 VxD 设计中采用 OnW32DeviceIoControl 来响应主控程序的调用。当设备驱动程序获取中断后，在 VhardwareInt 的成员函数 OnHardwareInt 中调用 PostMessage 函数，将消息 canbusMsg 发送到当前应用程序的窗口 canbusWindow，应用程序获取驱动程序发送的消息之后，就可以进行后续的数据处理了，其中的数据读写操作以及数据的预处理由前文所述的后台处理线程函数 CanbusWrRd 来完成。

3　结束语

本文重点阐述了采用 EPLD 以及双口 RAM 技术实现基于 CAN 总线的 PC－CAN 适配卡的设计及其完整实现。通过这一技术简化了系统的设计，缩小了系统的规模，提高了系统可靠性，解决常规适配卡设计中存在的一些问题，对现场总线的应用有一定的推广价值。

参考文献：

[1] 阳宪惠. 现场总线技术及其应用 [M]. 北京：清华大学出版社，1999.

[2] 邬宽明. CAN 总线原理和应用系统设计 [M]. 北京：北京航天航空大学出版社，1996.

[3] 薛钧义，张彦斌. MCS－51/96 单片微型计算机及其应用 [M]. 西安：西安交通大学出版社，1997.

[4] Tanenbaum A S. Computer Networks [M]. Prentice－Hall International Inc，1997.

（上接第 472 页）

参考文献：

[1] 何强，何英. MATLAB 扩展编程 [M]. 北京：清华大学出版社，2002.

[2] 杨行竣，迟惠生. 语音信号数字处理 [M]. 北京：电子工业出版社，1995.

[3] Lawrence Rabiner，Biing－Hwang Juang. 语音识别基本原理－影印版 [M]. 北京：清华大学出版社，1999.