摘　要

随着无线通讯和个人数字助理（Personal Digital Assistant, PDA）的迅猛发展，语音识别技术在嵌入式系统和设备上的应用逐渐成为新的热点。但是，受 PDA 设备的存储空间和运算速度等因素的制约，语音识别应用的普及受到了限制，为此需要有针对性地改进语音识别技术。本文基于嵌入式操作系统 Pocket PC，以非特定人语音命令识别的声控拨号系统（声控拨号器）为研究平台，研究在不影响识别性能的前提下降低声控拨号系统时空复杂度的方法。论文工作包括如下方面：

（1）研究了在不降低识别性能的前提下，选用合适的语音识别基元、选取较少特征维数进行声学建模的方法。通过实验比较选定扩展声韵母作为识别基元，并对 PC 上使用的多维特征进行压缩，提出一种把基于基元拼接、整词识别的声学建模技术用于嵌入式声控拨号系统的方案。

（2）研究了正确候选搜索路径在 Viterbi 束搜索中的排名顺序随输入语音时间（即输入特征序列帧号）之间的动态变化关系，以及排序后相邻排名的搜索路径分数差值随时间的变化关系，提出了一种结合搜索路径分数差值进行调整的动态调整直方图剪枝的搜索策略，减少了搜索解码的计算量，提高了速度。

（3）研究了搜索解码过程中的似然分重复计算问题，提出在 Viterbi 解码中利用速查表加速计算的方法，进一步提高搜索解码的速度。

（4）研究了传统端点检测技术在实际嵌入式应用中出现的零漂移、开头纯零等问题，提出端点检测算法的改进方法以提高端点检测的准确性。

在上述方法的基础上，设计并实现了一个非特定人的可定制词表的实用语音命令系统。在实际的 PDA 设备上的测试表明，当词表随机选取为 200 个中文人名时，识别正确率可达到 98.70%，相对于基于标准算法的系统，识别速度可提高约 80 倍，而搜索存储空间可节省约 30%。

关键词：语音识别　声控拨号　个人数字助理(PDA)　动态调整直方图剪枝

Abstract

With the rapid development of wireless communications and Personal Digit Assistant (PDA), the application of speech recognition technologies to the embedded systems and devices has become a hot spot in research. However, due to memory and speed limitations of PDA, it fails to deliver optimum performance and therefore needs further tailor-made improvement. To find a solution, we have conducted researches on speaker-independent speech command recognition system and voice dialing system, based on Pocket PC, an embedded system platform. With the aim to overcoming the constraints of memory space and computation speed under acceptable recognition performance degradation, we took the following steps to reduce the temporal and spatial complexity of voice dialing system:

(1) By studying how to choose appropriate acoustic recognition unit and how to do acoustic modeling with less dimensioned acoustic features under acceptable performance degradation, and by carrying out sufficient experimental comparisons, we decided to use Extended Initial/Final (XIF) as acoustic recognition unit. Additionally, through experiments, we found a more compact acoustic feature for acoustic modeling and developed a new acoustic modeling method based on acoustic recognition unit concatenation and full word recognition, which has proved suitable in embedded voice dialing system.

(2) By studying the dynamic interdependency between the rank of correct token path during Viterbi decoding and the input speech frames, i.e., input feature train frames, and the interrelationship between the difference scores of token paths and the input speech frames in Viterbi beam search, we proposed a new search strategy named Dynamically-Adjustable Histogram Pruning with the integration of the difference scores of token paths, which could significantly reduce the redundant computation and accelerate the search decoding.

(3) After studying the repetition of computing likelihood scores, we applied the likelihood score lookup table to further accelerate the Viterbi decoding.

(4) After studying the problems of zero drift and the beginning pure zero wave when using traditional end point detection technologies in the practical embedded applications, we made improvements accordingly to enhance the accuracy of Voice Active Detection (VAD).

Through sufficient experiments, we synthetically applied the methods proposed above in designing and implementing a practical speaker-independent, user definable voice dialing speech recognition system. The practical testing experiments in PDA device demonstrate: in a randomly chosen 200-Chinese-word vocabulary, its recognition accuracy rate reaches 98.70%. Furthermore, it could realize better recognition speed by 80 times and save decoding space by 30% in comparison to the baseline system using standard Viterbi decoding method.

Key Words:

Speech Recognition Voice Dialing Personal Digital Assistant (PDA) Dynamically-Adjustable Histogram Pruning

目 录

第1章　引言

1.1　嵌入式语音识别应用概述

随着计算机软硬件技术、半导体技术、电子技术、通讯技术和网络技术等的飞速发展，人类已经进入后 PC 时代[1]。而近年来，语音识别技术的应用分为两个发展方向：(1)是 PC 上的大词汇量连续语音识别系统，主要应用于计算机的听写机，以及与电话网或者互联网相结合的语音信息查询服务系统等；(2)是小型化、便携式语音产品的嵌入式语音识别应用，如无线手机上的拨号、汽车设备的语音控制、智能玩具、家电遥控等方面的应用。

嵌入式语音识别系统是指应用各种先进的微处理器在板卡级或是芯片级用软件或硬件实现语音识别技术[2]。实验室中高性能的大词汇量连续语音识别系统代表当今语音识别技术的先进水平[3]，但由于嵌入式平台在资源和速度方面的限制，其嵌入式实现尚不成熟，而中小词汇量的命令词语音识别系统由于算法相对简单，对资源的需求较小，并且识别率和鲁棒性较高，能满足大多数应用的要求，因而成为嵌入式应用的主要着眼点。

在无线通讯和个人数字助理 (Personal Digital Assistant, PDA) 迅猛发展的今天，嵌入式语音识别技术逐渐成为新的研究热点，国内外许多公司及研究机构，像 IBM、Microsoft、得意音通、言丰科技等，以及清华大学、中科院自动化所、中科院声学所等，都在开发自己的嵌入式语音识别引擎[4]。目前嵌入式语音识别应用主要从以下两个层面进行：

1) 针对 DSP 专用处理器芯片的应用。主要是基于芯片处理器加快语音识别的速度，比如清华大学语音技术中心的基于 ADSP 芯片的语音命令开发[5]，清华大学电子系的定点 DSP (Digital Signal Processing)芯片片上系统 SOC (Systems On Chips)实现[6][7]等，都是把语音识别算法固化到芯片上。在早期硬件资源不够发达的情况下，比较多地采用这种方法。不足之处是：这样的系统往往都是特定人识别系统，需要在线训练模型，运算精度不高，可支持的词表规模不大，变化不灵活，识别效果不够好。

2) 基于算法层面的应用。出发点是减少模型的规模和搜索空间，提高解码的速度；同时也可以牺牲运算精度，增加软件层面的定点计算，进一步提高运

算速度[8][9]。随着嵌入式设备硬件资源的不断发展，计算和存储能力的提高，这种方式的应用越来越多。它既可以在线训练模型，实现特定人语音识别，也可以事先载入训练好的模型，实现非特定人语音识别，具有识别精度高，应用灵活的特点。不足之处是：声学模型需要在规模和精度之间折衷；还需要严格控制识别过程中搜索空间的消耗以及提高搜索速度。

本文将侧重从算法层面考虑对嵌入式语音命令识别应用进行改进，希望能够把在 PC 上应用比较成熟的语音命令识别技术，经过针对嵌入式设备和应用环境的特点进行研究并改进后，应用到嵌入式系统 PDA 上。

1.2 嵌入式语音识别应用的关键技术

一般来讲，和传统的语音识别系统一样，嵌入式语音识别应用的关键技术主要集中在如下几个方面[10]：

(1) 声学特征。特征提取与选择是语音识别的一个重要环节。特征提取解决了时域语音信号的数字表示问题，而特征选择则通过选取有效的特征为模式划分提供依据。特征提取与选择的好坏直接影响到识别器的性能。目前在 PC 上普遍采用的多维特征导致声学模型规模的增大，对于嵌入式环境来说是一个较大的负担，需要寻找合适的声学特征。

(2) 声学模型。HMM 能描述不同层次的语音单元，由 Viterbi 解码算法[11]可以得到与语音序列对应的最佳状态序列，便于解决连续语音识别的问题。到目前为止，HMM 模型及其改进模型仍然是语音识别系统中声学模型的主流选择。在嵌入式语音识别系统中，如何选择合适的声学模型，既有较高的识别率，又有较小的规模，是声学建模的重点问题。

(3) 搜索算法。语音识别中的搜索，就是寻找某个最佳状态序列来描述输入语音信号，从而得到语音信号的解码序列。而搜索的依据是语音信号在声学模型的打分以及加入语言模型的概率。针对 HMM 模型基本的搜索策略为 Viterbi 解码算法和帧同步算法[12]，但是存在的问题是搜索路径会随着时间的增长而急剧膨胀，因此如何采用合适的剪枝策略来减少搜索的空间消耗并提高搜索的速度是嵌入式语音识别的难点之一。

(4) 自适应与鲁棒性问题。由于在实际应用中存在不同的说话人、说话方式、环境噪声、传输信道等因素，语音识别系统的性能会急剧下降。所以必须保证

在不同应用环境下性能的稳定。解决的办法可以分为基于语音特征的方法和基于模型调整的方法。前者的目标是寻找更好的、高鲁棒性的特征参数。后者的目标是利用少量的自适应语料来修正或变换原有的说话人无关模型，使其成为说话人自适应模型。

对于具体的嵌入式系统 PDA 来讲，如前所述，PC 上中高性能的非特定人大词汇量连续语音识别在实验室已经取得令人满意的效果，但是在 PDA 这样的嵌入式移动设备上，由于以下原因还无法发挥其性能[4]：

(1) 运算速度不高。处理器一般运行的是 MIPS 指令，处理能力小，只相当于 PC486 水平。

(2) 存储空间不大。存储卡一般只有 16M 或 32M 大小，与目前 PC 的内存大小相差若干个数量级。

(3) 声音信道不同。内置麦克风信噪比比较低，声音信道不同于普通的 PC 或固定电话信道，而且目前用以训练声学模型的数据库大多不是 PDA 设备环境下的。

因此，在 PDA 上进行嵌入式语音命令识别应用，其关键技术是要求算法在保证识别效果的前提下，减少系统的时空复杂度，以适应嵌入式平台存储资源少、实时性要求高的特点[13]。如何在实际应用中采用合适的解决方法，更是关键。基于此，如何从声学建模和搜索策略进行改进和应用亦是难点：在不特别降低识别性能的情况下如何进行小规模而高精度的声学建模；如何减少搜索解码的存储占用空间，提高解码的效率和正确率，提高搜索解码的速度；最终提高整个声控拨号系统的性能，使之得以实用。

1.3 本文的主要工作和论文安排

本文的主要工作是以嵌入式操作系统系统 Pocket PC 为实验平台研究基于非特定人语音命令识别的声控拨号系统（声控拨号器）。针对在 PDA 上的存储空间和运算速度的限制条件，在不影响识别率的基础上，降低声控拨号系统的时空复杂度，采取了如下措施：

(1) 研究了在不降低识别性能的情况下，如何选用较少的特征维数进行声学建模，结合扩展声韵母作为识别基元，提出了一种适合嵌入式设备上声控拨号器的声学建模方法。

(2) 研究了正确候选搜索路径在 Viterbi 束搜索中的排名顺序与输入语音帧数的动态变化关系，研究了排序后相邻排名的搜索路径分数差值随时间变化的关系，进而提出了一种结合搜索路径分数差值动态调整直方图剪枝的搜索策略。

(3) 研究了搜索解码过程中的似然分重复计算问题，提出在 Viterbi 解码中利用速查表加速计算、提高解码速度的方法。

经过实验研究，本文综合应用上述方法以及其他实现技巧，设计并实现了一个非特定人的可定制词表的语音命令识别系统—“得意声控联系人”，改进了系统的性能，并将其实用化和推广。

接下来的章节将按照如下方式组织：首先介绍了适用于嵌入式声控拨号的声学模型基元的选取依据、以及基元拼接整词识别的建模方法；其次重点介绍了在声控拨号系统中针对嵌入式环境所采用的结合搜索路径分数差值动态调整直方图剪枝的搜索策略、以及利用速查表优化搜索的方法；紧接着介绍声控拨号器系统中所采用端点检测技术极其改进；然后总体介绍了综合以上方法的应用—非特定人的可定制词表的声控拨号器，给出了实验结果并给出结论；然后是本文的工作总结与未来工作展望；在论文的最后，给出了本论文所引用的参考文献。

第2章　嵌入式系统上的声学建模技术

声学建模是语音识别中声学层面处理的关键步骤。声学模型用来描述识别基元对应的特征矢量序列的产生过程。通过声学建模，可以估计待识别特征矢量序列所对应的语音识别基元，从而完成特征矢量序列到语音识别基元的识别转换。因而，在语音识别系统中，声学模型是整个系统的核心部分之一，声学模型的性能决定着整个语音识别系统的整体性能。本章将介绍针对嵌入式声控拨号系统的声学建模过程，包括模型类型、识别基元、声学特征的分析和实验比较选定等。

2.1　声学模型在语音识别系统中的作用

一般的语音识别系统中的核心部分包括声学模型和语言模型。二者之间的关系可以用图 2-1 来描述[10]：

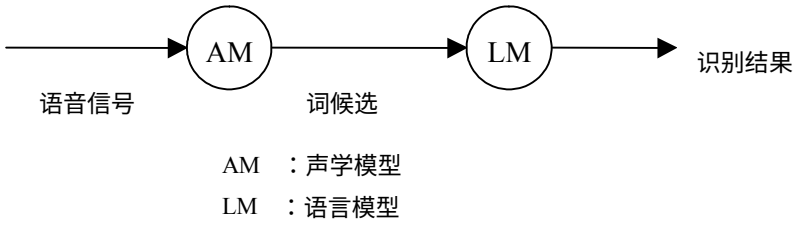

图 2-1　语音识别系统的基本组成

我们可以看到，声学模型是语音识别系统中的枢纽，直接承担着将语音特征数据识别为声学模型串的重任。因此，声学模型的好坏决定了整个语音识别系统的整体性能。对于语音命令识别系统来说，由于系统构成比较简单，往往为孤立词识别系统，一般不需要语言模型的参与，那么核心部分就只剩下声学模型，因而声学模型的重要性更加明显。

2.2　声学建模中模型类型的选择

声学模型主要功能是对识别对象进行模式划分。进行模式划分的方法很多，

比如基于概率统计模型的方法和基于人工神经网络的方法，具体对于语音识别来讲，基于概率统计模型的方法，因为使用方便、准确有效，得到了比较广泛的应用，在语音识别领域中占据了重要的地位。而在基于概率统计模型的模式划分方法中，隐式马尔可夫模型(Hidden Markov Model, HMM)比较符合语音信号作为随机信号的变化规律，对语音信号的描述比较准确，具有较高的识别率，所以 HMM 及其变形已经成为语音识别系统中应用最多的方法。

2.2.1 HMM模型及其变形的种类

一般来讲，HMM 模型及其变形主要有如下几种类型：

根据观察输出概率矩阵中是基于离散矢量、连续密度或者是二者的综合，HMM 分为离散 HMM 模型(Discrete HMM, DHMM)、连续 HMM 模型(Continuous HMM, CHMM)和半连续 HMM 模型(Semi-Continuous HMM, SCHMM)[14]。

针对经典 HMM 时空复杂度高，尤其是 Viterbi 解码过程的复杂性，人们提出了一个基于高斯混合分布和观察序列分段的声学模型，即混合高斯连续概率模型（Mixed Gaussian Continuous Probability Model, MGCPM）[15][16] 。

后来又根据分段概率模型，人们也研究变换出新的基于中心距离概率模型，实现了对 HMM 模型的简化和创新，就是 CDCPM 模型 (Center-Distance Continuous Probability Model, CDCPM)[17]。

基本上来讲，后面两种 HMM 的变形改进主要针对减少或简化连续 HMM 模型的规模或复杂度，因而不可避免会影响到连续 HMM 模型的识别精度。

2.2.2 HMM模型的选择

在相同条件下，比如相同训练数据、同等模型规模，HMM 的三种概率密度模型的建模精度按照离散的、半连续的、连续的顺序递增。离散参数的 HMM(DHMM)虽然在减小模型规模上很有成效，但是不可避免地会引入量化误差；而半连续的 HMM(SCHMM)的分布假设与实际分布的差异也引入了模型不匹配的问题；连续 HMM(CHMM)的精度最好，但是需要比较大的模型规模。所以如果模型规模的限制条件比较严格，为了追求最低的模型存储，而对精度要求不高的话，离散 HMM 在早期是比较合适的选择；如果对精度要求很高，而对模型存储规模限制要求不高的话，无疑高混合数的连续 HMM 模型是最好的选择；如果既要求模型精度比较高，对模型的规模也有一定限制，那么较低混

合数的连续 HMM 模型或者较高混合数的半连续 HMM 可以考虑。

综合考虑到国内外现今语音识别的主流方法，连续 HMM 模型具有识别精度高的优势，应该是声学模型的首选，我们的目标是如何进一步将连续 HMM 模型的规模降低下来，达到适合嵌入式设备的应用条件。在后续实验研究中我们发现，在特定的语音命令识别条件下，通过选择合适的识别基元，以及通过合适的声学特征挑选，可以进一步减小连续 HMM 声学模型的规模，而对于识别的精度正确率只有可接受的轻微下降。

2.3 声学建模中识别基元的选择

语音识别基元的选择在语音识别尤其是语音命令识别中是重要的环节。一般来讲，识别基元的选择应该基于如下两个原则[18]：

(1) 具有灵活性，即灵活的可组合性能，能够代表语音中的比较独立的一些个性，可以组成其他的语音单位；

(2) 具有稳定性，即它应该使得语音中的共性能够得到相当的体现，从而保证识别基元对不同环境的适应能力（即鲁棒性）。

在这两个原则中，灵活性希望基元尽可能地小，如音素、声韵等；而稳定性则希望基元尽可能地大，如词、甚至词组。在实际应用中，这两个方面需要综合考虑。

2.3.1 声学建模中识别基元的种类

对于西方的语言，常常采用音素(Phoneme) 或上下文相关音素(Triphone)作为识别基元。其中 Triphone 模型由于考虑了上下文相关的影响，极大地提高了语音识别系统的性能，在实际系统中取得了很好的效果，是西方语言的语音识别系统的主流发展方向[19]。

由于汉语比较特殊，它是由音节组成的语言，所以除了能够仿照西方语言采用音素作为识别基元之外，可以采用音节作为汉语语音识别基元。此外，每个音节由声母和韵母组成，声韵母作为识别基元也是近年来一种不错的选择。

所以，针对汉语语音识别系统，我们可供选择的几种子词识别基元种类及其特点如下：

音节(Syllable)基元：汉语的音节结构固定，每个音节对应一个汉字，训练

数据的标注比较容易。采用音节作为识别基元的好处是利用了汉语语音的特点，能取得好的识别效果。实验证明当进行上下文无关建模时，音节作为识别基元能取得很好的识别效果[18]。缺点是音节本身的数目较大（汉语共有 418 个无调音节），不利于上下文相关的建模。此外音节模型还存在训练不充分的问题。

音素(Phoneme)基元：音素是语音的最小单位，一共有 35 个，其中辅音基元 22 个，元音基元 13 个。其最大的优点是数量很少，在训练数据相对固定的情况下，它们可以得到充分的训练。此外，在改换任务或建立新词条时无需重新建模和训练，甚至在跨语言的识别系统中也能得到很好的应用。但是由于协同发音(Co-articulation)现象的存在，采用上下文无关的音素模型有较大的误识率。改进的方法是进行上下文相关(Context Dependent, CD)的音素建模，也就是人们通常称作的 Triphone 模型，但是容易造成模型规模太大。

声韵(Initial/Final, IF)基元：这是根据汉语的特点建立的模型。汉语中的汉字是单音节的，而汉语中的音节是声韵结构的，这种独特而规则的结构，使对音节、以及词条的表示变得比较规则和统一。其好处是声韵母的数目比较少，一般讲，声韵母共有 59 个，其中声母 21 个，韵母 38 个；另外，声韵结构比较稳定，充分利用了汉语语音的特点。上下文无关的声韵母模型具有规模小，速度快的优点，但是识别率相对于音节模型比较低[20]。为了提高系统的识别率，需要进行上下文相关的声韵母(Context Dependent Initial/Final, CD-IF)建模。缺点是只适用于汉语语音识别系统。

扩展声韵(Extended Initial/Final, XIF)基元：在声韵基元 IF 的基础上，为了使声韵结构更加结构化，又提出扩展声韵母集[21]，与标准的声韵母定义相比，增加了 6 个零声母{_a, _o, _e, _I, _u, _v }，这样，每个音节都是由两部分组成的，分别对应其声母部分和韵母部分。当使用标准的声韵母基元集合时，有一些音节只有韵母部分，而没有声母部分。所以，考虑上下文相关信息时，这些韵母既可以搭配声母，又可以搭配韵母，因此，上下文相关声韵母基元数目会很大，超过 10 万个。而使用扩展的声韵母基元集合时，上下文关系比较确定，基元数目减少为约 3 万多个，有效地缓解了数据稀疏问题。同时，使用扩展声韵基元也有效地减少识别中的插入错误，其性能也优于标准声韵母基元。扩展声韵基元的缺点也是只适用于汉语语音识别系统。

对于以上这些可供选择的子词识别基元，在孤立词识别的语音命令识别系统中，我们需要经过实验从识别率、速度、存储规模，以及实际实现可行性等

因素上进行比较和选择。

2.3.2 基元拼接整词识别

目前在非特定人孤立词系统中普遍采用两种方法：一种是整词建模整词匹配的方法；一种是基于词树搜索[22]和子词建模[23]的方法。

第一种方法把整词作为建模和识别单元，在训练的时候以整个词为单位训练出特定的词模型，而且词模型的状态数目与词所包含的音子的数目基本保持一致。整词建模方法的优点是：计算量小，识别率高；缺点是：词表必须固定，不能及时变化，或者必须在线训练新的模型，而且需要采集大量的训练语音，这给实际应用带来了不便。

第二种基于词树搜索和子词建模的算法，在训练的时候以子词为单位训练出子词模型，而在识别时则采用基于词树的一遍或多遍路径搜索算法。该算法的优点是：建模精度高，搜索空间小，计算量也比较小，可以适用于较大规模的词表，而且使用时无需在线训练，更改词表方便，有利于实际应用；缺点是：算法复杂，需要的模型存储空间大，而且识别时会引入搜索误差，从而引起识别率的下降。

针对实际情况，为了达到：无需在线训练、可定制词表、词表更改灵活等要求，并且对速度和识别率都有很高的要求，必须采用比词更小单位的子词为基元来建模。本文综合从整词建模方法和基于词树搜索、子词建模的方法中获得启发，提出基元拼接整词识别的建模方法，选择子词作为模型基元，离线事先训练好基元模型，在系统中应用的时候，根据词表中每个词的子词基元序列拼接成整词模型进行识别，可以方便地定制各种词表。

2.3.3 识别基元的实验选定

在确定了使用子词为基元来进行基元拼接整词识别的建模方法后，为了最终确定具体使用何种子词基元（音素、声韵、扩展声韵、音节）作为最终的识别基元，本文对几种类型的识别基元在识别精度和模型规模方面进行了综合比较实验。

本文中使用的数据库是“ 863 数据库 ”中的男声数据库[24] 。数据库中的句子是用略带口音的普通话读出的。数据库中共有 1,560 个不同的句子，被划分为三组，分别称为 A，B 和 C 组。库中共有 80 个人的语音数据。全部语音数据

的文本信息以及音节一级的标注信息都是已知的。标注信息的获得是利用手工标注和机器切分相结合的方法得到的。

本文中从数据库中选取 70 人的数据作为训练集合，剩余 10 人数据作为测试集合。所以，测试集合中的说话人都不在训练集合中。

实验中使用 42 维的 MFCC（Mel-Frequency Cepstrum Coefficients）作为特征参数，包含能量参数，以及一阶差分和二阶差分参数。

本文使用 HTK v2.2 工具进行模型训练[25]。测试结果用连续识别的音节正确率来进行评价。

其中，各种基元类型的音节识别正确率的结果如图 2-2 所示。

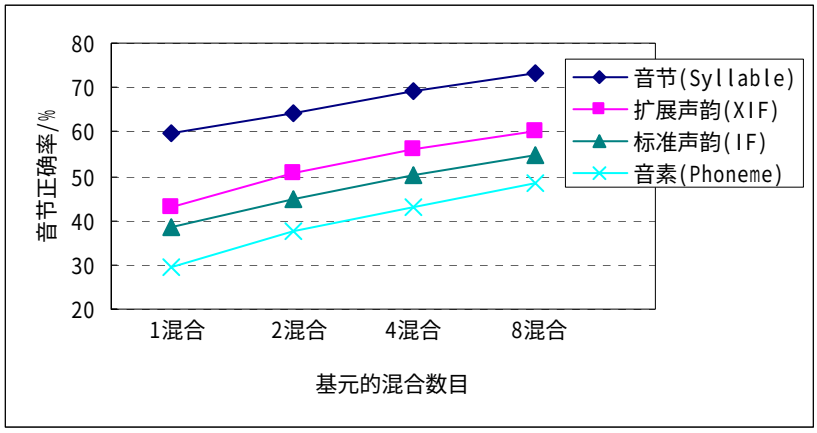

图 2-2　各种基元类型识别精度比较

而各基元类型对应的模型规模如表 2-1 所示。

表 2-1　各种基元类型模型规模比较

基元类型	模型规模/KB			
	1 混合	2 混合	4 混合	8 混合
音素（Phoneme）	43	83	158	231
标准声韵（IF）	74	142	272	395
扩展声韵（XIF）	82	156	299	435
音节（Syllable）	1039	1976	3787	5510

综合以上图 2-2 和表 2-1 的实验结果可以发现，虽然音节基元(Syllable)的识别精度比较好，但是由于模型规模太大，无法适应嵌入式系统存储资源少的特

点，因而并不适用；而音素基元(Phoneme)虽然模型规模小，但是相对于其它基元类型，其识别精度最差，因而也不适用；至于标准声韵基元(IF)和扩展声韵(XIF)，他们在模型规模上比较小，识别精度较高，两者的模型规模差别也比较小，但是识别精度差别还是很大的，很明显扩展声韵基元(XIF)要优于标准声韵基元(IF)。综合以上考虑，本文选择了扩展声韵基元(XIF)作为识别基元，而且通过后续对声学特征的研究，针对特定的语音命令识别的应用条件，进行了特征的压缩选择，可以进一步减小模型的规模，而在识别精度上没有大的影响，具体见后续章节。

2.4 声学建模中特征的选择

语音特征刻画了声音信号的共性和特性，针对不同的应用情况，选择合适的声学特征对于整个语音识别系统的性能有很大的影响。

2.4.1 声学特征的种类

语音信号的特征主要有时域特征参数和频域特征参数两种。其中，时域特征参数中用得比较多的有：短时平均能量、短时平均过零率、共振峰和基音周期等。频域特征参数用得比较多的主要有傅里叶频谱，比如基于线性预测分析(Linear Predictive Coefficient, LPC)的倒谱系数参数即(Linear Predictive Cepstrum Coefficient, LPCC)，以及基于 Mel 频率弯折的倒谱系数参数(Mel-Frequency Cepstrum Coefficient, MFCC)[26]。如果把时域特征参数和频域特征参数结合起来，就形成了时频特征参数，这种参数的分析需要用到比较复杂的时频分析技术[27]。

2.4.2 声学特征的实验选定

一般来讲，在特征方面，Mel 倒谱系数 MFCC 符合人耳的听觉特性，能够比较正确的刻画语音的特征，而且被证明抗噪性能较好。在嵌入式语音识别应用中，嵌入式设备的使用环境的是多样的，经常有各种各样的环境噪声，所以要求声学模型的特征必须具有较好的抗噪性能，无疑，Mel 倒谱系数 MFCC 更加适合于嵌入式语音识别应用。

在 PC 上应用连续语音识别的时候，现在主流采用的特征是 42 维特征({13 维 MFCC+1 维能量}+一阶差分+二阶差分)[28]，但是采用这么多的特征维数，导

致了模型规模的急剧增大，如果采用该 42 维特征的声学模型，其模型存储规模并不适合嵌入式设备紧张的存储资源，因此在不影响识别率的情况下必须对特征维数进行缩减。

首先，以往的实验证明，在特征中去掉二阶差分系数后，比如({13 维 MFCC+1 维能量}+一阶差分)，对于识别率的降低是很轻微的，即对于模型精度影响不大，这个也可以在后续的实验中得到证明，但是由此带来的是模型存储规模降低了 33.3%。

其次，考虑到语音频带中低频携带噪音成分较多、高频成分则携带了基音和谐波特性，对于非特定人系统的语音特征影响不大，为了适应嵌入式设备上存储资源有限的特点，本文在提取 MFCC 特征时，通过滤波器组过滤掉语音频带的低频和高频成分，从而将原先的{13 维 MFCC+1 维能量}缩减为{9 维 MFCC+1 维能量}，并忽略二阶差分系数，最后经过倒谱均值归一化(Cepstrum Mean Normalization, CMN)去噪处理，得到“ {9 维 MFCC+1 维能量}+一阶差分 ”一共 20 维特征参数。

为了测试不同特征的性能，我们进行了实验比较，进而进行了声学特征的选定。

我们用该最终压缩后得到的 20 维特征({9 维 MFCC+1 维能量}+一阶差分)与原先的 42 维特征({13 维 MFCC+1 维能量}+一阶差分+二阶差分)和初步压缩的 28 维特征({13 维 MFCC+1 维能量}+一阶差分)针对声控拨号器（孤立词识别系统）进行了实验比较。

声学模型训练实验条件如下：声学模型采用了连续 HMM 模型；基元采用扩展声韵基元；训练数据情况为：15 人 863 数据库男声数据，15 人 863 数据库女声数据，150 人电话数据，总共约 30,000 句；训练数据均为 8K 采样率，16 位采样深度。模型的训练采用 HTK 工具进行，分别得到三种特征的对应声学模型。

测试词表分别为随机中文人名，大小分别为 200 词、300 词、600 词（词长为 2 个字或 3 个字），测试数据是在 Dopod 掌上电脑（采用 Pocket PC 2002 操作系统和 StrongARM 芯片）上通过 8K/16bit 采样实地得到的，由 6 个人（4 男 2 女），每人说一遍词表中的人名，每个声音录制长度约为 3 秒，采用标准 Viterbi 进行孤立词解码，得到如图 2-3 所示的实验结果。

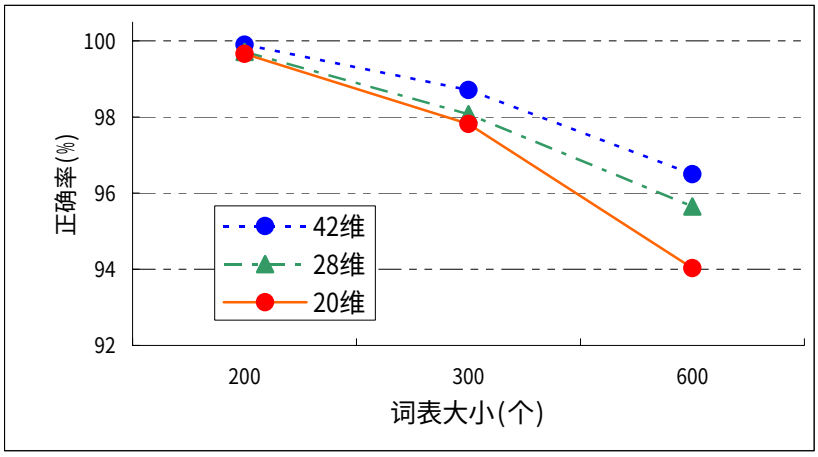

图 2-3　不同特征维数模型识别结果比较

实验证明，在中小词汇量的语音命令识别系统中，采用这样的 20 维特征相对于 42 维特征的识别率降低是可以接受的；而且，这样做不但能够减小模型的规模（因为语音特征向量的维数小），也可以减少后续识别搜索过程中似然分的计算时间，从而提高系统的识别速度，因而更具有现实意义。

2.5 本章小结

本章结合嵌入式语音命令识别系统应用的特点和限制条件，对声学建模过程的各种情况进行比较研究，包括对 HMM 模型的类型进行分析比较、对各种识别基元的优缺点进行分析取舍、对多种声学特征进行实验比较选择，还包括对声学模型的训练进行训练数据的挑选，经过实际声学建模的实验比较，研究了在不特别降低识别性能的情况下，如何选用较少的特征维数进行声学建模，结合扩展声韵基元作为识别基元，提出了一种适合嵌入式设备上声控拨号器的基元拼接整词识别的声学建模方法。该声学建模最终的选择是：识别基元采用扩展声韵 XIF 基元；声学模型采用连续 HMM 模型；声学特征采用 20 维 MFCC 特征({9 维 MFCC+1 维能量}+一阶差分)。本章提出的声学建模方法将在后续章节中被采用并进行进一步的研究和应用。

第3章　动态调整直方图剪枝策略

3.1　搜索解码综述

一般来讲，一个语音识别系统由几个主要的模块组成，包括特征提取、声学模型、语言模型和搜索算法等，每一个模块都起了非常重要的作用。从图 3-1 可以看出：搜索算法模块处于整个系统的核心部分，声学模型知识、语言模型知识和其余各种有帮助的知识源都要通过搜索算法来起作用，搜索算法的优劣直接关系着识别系统的性能和效率，是语音识别中最关键的技术之一。本章将主要介绍针对语音命令识别系统进行的搜索策略的研究与实现工作。

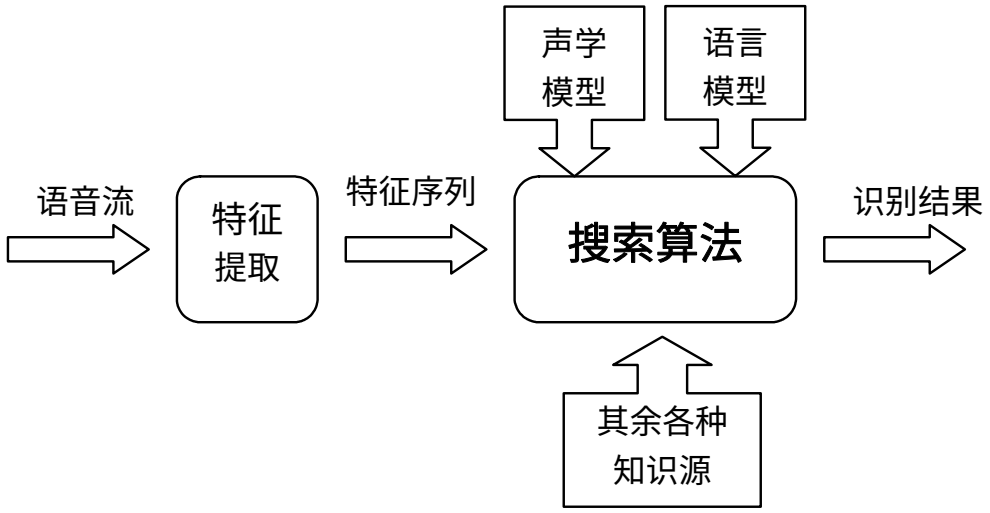

图 3-1　搜索算法在语音识别系统中的作用

3.1.1　搜索策略需解决的问题

在连续语音识别中，搜索策略主要需要解决下面三个问题[29]：

(1) 准确性：有效地利用各种知识源，使识别结果尽可能的准确。

(2) 高效率：尽量快地得出识别结果，理想情况是语音流一经输入立刻得到识别的文本结果。

(3) 低消耗：尽量少地占用系统资源，包括需要的内存空间，硬盘空间等。

是否能够很好的解决这三个关键问题也是评价各种搜索算法优劣的标准。

对于搜索的效率和资源消耗，我们认为应有一个最低的限度，不能以牺牲大量的识别精度来谋求实时，同时也不能把需要数十倍甚至数百倍于现有计算能力的算法称为合适的算法。

虽然传统上还是使用识别准确率作为评价搜索算法优劣的主要标准，但随着语音识别技术逐渐地走向市场，搜索算法的效率和资源消耗两项评价指标也越来越重要，因为它直接关系到整个语音识别系统的效率和占用的资源。因此搜索算法研究的重点也由如何提高搜索算法的准确性转移到在不损失现有识别准确性的前提下，如何使搜索算法实时化和具有较低的资源消耗。

3.1.2 搜索算法分类

目前的搜索算法从基本原理上可以分为两类：时间不同步的基于启发函数的深度优先的算法（或称为 A*算法），和时间同步的宽度优先算法（或称为时间（帧）同步 Viterbi 搜索）。

(1) 时间不同步 A*搜索算法

时间不同步的 A*算法可以描述如下[30][31]：在算法的每一步迭代过程中，有一个排序了的列表，该列表中每一个元素记录着一条局部路径和该条路径的启发式得分。选择得分最高的路径按照词法树进行扩展并计算扩展部分的似然得分和重新估算启发函数得分，把似然得分和启发函数得分相加，即为该条路径的评估函数得分，并按照评估函数得分高低插入列表中。评估函数描述了该条局部路径扩展完整后的最大可能概率得分，即为整条路径概率得分的上限估算。当出现在列表头上的路径是一条完整的词时，得到识别结果，算法结束。只要搜索过程的启发函数满足一定的条件，以上算法保证能找出最优的路径，并可以非常容易地得到任意个候选。

假设路径 f（完整或不完整）在时间 t 时得分为 $\alpha_t(f)$，其未扩展部分的启发函数得分估算为 $\beta_t^*(f)$，则路径 f 的评估函数定义为：

$$h^*(f)=\max_{0\le t\le T}\alpha_t(f)*\beta_t^*(f) \qquad \text{(式 3-1)}$$

搜索的目标是找出一条完整的路径其概率得分 $h(f)$最大。我们说启发函数是可采纳的(Admissible)，当且仅当 $h(f)\le h^*(f)$，其中 $h(f)$为任何一条路径 f 扩展后的概率，等号当 f 为完整的路径时成立。当选择的启发函数满足可采纳的条件时，能够保证识别结果按照概率得分从大到小的次序产生。

在 A*搜索过程中，由于堆栈深度的几何级扩展，我们也需要对堆栈进行剪枝，例如只取前固定个数的局部路径进行进一步的扩展，也可采用束搜索(Beam Search)的办法，取一个门限与最优路径进行比较以决定取舍。

(2) 时间（帧）同步 Viterbi 搜索算法

时间同步 Viterbi 搜索算法是以动态规划技术为基石。1989 年 ChinHui Lee 在前人工作基础上，在特定的搜索空间中，按节点间词的连接方式，对连续语音识别的搜索问题进行了深入的研究，发表了对帧同步网络搜索算法的详细研究报告[32]。帧同步网络搜索算法的理论基础大多来自于 Ney. H 的一遍动态规划算法。帧同步搜索算法首先设计一个有限状态网(FSN)，该 FSN 由节点和连接弧组成，分别代表一定的声学、语言学和语法知识以及相互的作用。语音识别的任务就是在 FSN 中搜索一定的最优路径。基于 Bellman 原理，即把一个全局最优问题的求解分解为小的局部问题并且形成递归联系，可以使用一个动态规划过程来得到最优路径，在语音识别搜索过程里，这个动态规划过程即由 Viterbi 算法代替。如下表示：

$$\begin{aligned}\hat{w}_1^N &= \arg\max_{w_1^N}\left\{\ P(w_1^N)\cdot\sum_{s_1^T}P(x_1^T,s_1^T\mid w_1^N)\ \right\}\\ &= \arg\max_{w_1^N}\left\{\ P(w_1^N)\cdot\max_{s_1^T}P(x_1^T,s_1^T\mid w_1^N)\ \right\}\end{aligned} \qquad \text{(式 3-2)}$$

其中：

$$\begin{aligned}p(x_1^T,s_1^T\mid w_1^N) &= \prod_{t=1}^{T}p(x_t,s_t\mid s_{t-1},w_1^N)\\ &= \prod_{t=1}^{T}[p(s_t\mid s_{t-1},w_1^N)\cdot p(x_t\mid s_t)]\end{aligned} \qquad \text{(式 3-3)}$$

在上面的求最佳词序列的过程中，使用了最大近似，即所谓的 Viterbi 近似[33]。在这种近似方法下，其 FSN 可以认为是一个双层网，这两层网络具有截然不同的性质。网络的上层起到了一个语法约束的作用，其节点表示语法单元的边界，弧表示语法单元及其转移。网络越简单，表明语法约束越松弛；反之，表明更强的语法约束。网络的下层是对上层网络的弧的展开，正是这样一个双层网络为帧同步搜索算法提供了搜索空间。

在查找局部最优子路径的过程中，大部分子路径的得分非常低。显然，完全保留这些得分较低的子路径在空间和时间上是不现实的，也是不必要的。束搜索(Beam Search)就很好地解决了这个问题，它在帧同步搜索过程中动态地对路径进行剪枝，抛弃希望不大的子路径，从而大大减少计算量。

当前最优子路径表示为：

$$W_{Max} = \arg\max_{w_1^N} \left(\Pr\left(\hat{W}_1^n \right) \right) \quad \text{(式 3-4)}$$

我们可以通过设定一个门限 f_{Beam}，保留所有得分在 $\Pr(W_{Max})$ 与 $f_{Beam} * \Pr(W_{Max})$ 之间的子路径进行下一步扩展，删除其余子路径。这样束搜索的搜索量将大大减少。束搜索是一个次最优的搜索算法，如果剪枝过于紧凑的话，在搜索过程中，可能会由于总体最优路径的子路径的某一个状态节点上得分太低，在搜索过程中被抛弃掉，从而永远不可能发现全局最优的路径。不过在语音识别中，这种次最优的算法往往是成功的[34]。

3.2 剪枝策略

在语音命令识别嵌入式应用系统中，对于实时性的较高要求使得时间不同步的 A*搜索算法并不适用，在实际中一般采用帧同步 Viterbi 搜索算法的改进算法，比如各种基于束搜索的剪枝算法。

大量的实验研究表明，在语音识别系统中，搜索解码的计算量占整个语音识别计算量的 80%以上[35]，所以提高搜索解码的速度和效率对于整个语音识别系统性能的提高具有决定性的意义。一般地说，基于 HMM 的搜索采用时间同步宽度优先的 Viterbi 方法[36]，会占用很大的搜索空间，搜索计算量也很大，显然不适合嵌入式设备，尤其当词汇量很大时，问题更为突出。最好的策略是采用剪枝技术[37]将计算量集中在最有可能的一些搜索路径上。

3.2.1 传统剪枝策略

传统的剪枝算法可以分为基于分数阈值剪枝和基于排名阈值剪枝。前者一般称为束剪枝(Beam Pruning)，后者一般称为直方图剪枝(Histogram Pruning)[38]。

基于分数阈值剪枝算法在每一时刻只保留似然概率分数最大的前若干条搜索路径，其中该预先设定的剪枝分数阈值称为束宽度。通常情况下，束宽度在

剪枝过程中固定不变，这种剪枝方法也称为固定束宽度剪枝。固定束宽度剪枝方法在阈值比较小的时候可以大幅度减少搜索空间，但当大多数路径的分数都非常接近的时候，则会导致搜索路径数急剧增加。所以该方法对于搜索存储空间的控制并不够有效，因为单纯从分数阈值上无法预知可以保留的搜索路径个数。

基于排名的直方图剪枝可以有效地避免类似情况。直方图剪枝在任何情况下都只保留固定个数的、分数排名靠前的候选者，可以防止分数阈值剪枝中出现的搜索路径空间的无法预知性。其中该预先设定的固定排名个数称为直方图剪枝的剪枝宽度。对于在嵌入式系统上的语音命令识别系统（孤立词识别系统）来说，采用基于排名的直方图剪枝方法可能是比较好的选择。

3.2.2 剪枝的层次

搜索剪枝在实际实现中通常可以在两个层面进行：状态级剪枝和模型级剪枝[39]。

在状态级上进行束搜索剪枝的时候，以声学模型的状态为单位对所有的状态序列的路径进行似然分数比较并判断是否保留或剪枝，而且同一个模型内的状态序列路径是各自独立地进行剪枝的。

在模型级进行束搜索剪枝的时候，则是以声学模型为单位对所有的基元模型序列的路径进行分数判断，以决定是否保留或剪枝。如果某个基元模型序列路径决定被剪掉，那么该模型里的所有状态序列路径也都将被剪掉。当然，模型级的束搜索剪枝在进行路径扩展的时候，也是以状态序列为单位进行打分和路径扩展的，只是在剪枝的时候，如果两条不同状态序列路径的终点都在同一个声学基元模型内，那么他们的关系是相关的，或者一起被保留，或者一起被剪掉。

相对而言，模型级的剪枝是在状态级的剪枝基础上进行比较粗放的判断剪枝。由于模型级的剪枝在存储上以及实现上都比较高效，而且计算量更节省，同时对最终的识别正确率的影响是很轻微的，因此更适用于嵌入式系统上的识别搜索。

3.3 动态调整直方图剪枝

基于排名的直方图剪枝对于搜索存储空间具有良好的可控制性和可预测性，因而比较适合应用于嵌入式环境下的语音命令识别应用。但是，基于排名的直方图剪枝和一般的剪枝算法一样，不恰当的剪枝也有可能把最优的路径过早地剪除，从而影响系统的识别正确率。比如，固定的剪枝宽度不能动态适应实际的情况，反而不利于在正确率和速度之间找到合适的平衡；要想获得的高正确率，需要增大剪枝宽度，这势必会导致计算量很大，因此需要进行改进。

3.3.1 正确候选搜索路径的排名趋势

一般说来，在搜索过程中，当剪枝宽度太小时，可能会由于全局最优路径的子路径在某一个状态节点上得分太低，导致在搜索过程中被抛弃掉，从而永远不可能发现全局路径最优，使得识别正确率过低；当加大剪枝宽度后，直方图剪枝的识别正确率随着提高，但是当剪枝宽度增加到一定程度后，识别率会呈现饱和的状态，其上限就是标准 Viterbi 剪枝方法的识别率。因此，采用剪枝算法的语音识别系统的识别结果的好坏和剪枝的宽度大小有很大的关系。本文希望通过研究正确候选的路径（全局最优路径）在搜索解码过程中的排名变化趋势，进而发现并利用其中的规律，避免过早将全局最优路径剪除掉，同时希望能够尽量多地将不能构成全局最优的路径剪除掉，以节省不必要的计算量。

本文对 200 个随机中文人名（150 个三字词+50 个两字词）构成的词表进行了标准 Viterbi 方法的测试，对于每一个测试语音数据，记录下正确候选的路径在每一帧输入语音的排名，经过统计得到如图 3-2 所示的含有正确候选的路径的排名趋势图，其中曲线表示的是平均数值，t_0 点表示的是排名曲线峰值点。

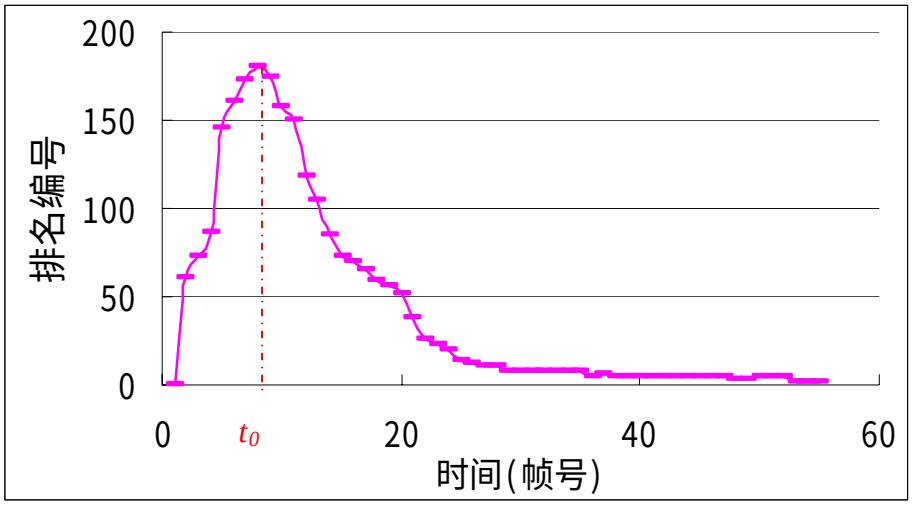

图 3-2　搜索路径中含有正确候选的路径的排名趋势图

从图 3-2 中可以发现，标准 Viterbi 解码过程中含正确候选的路径排名随时间（语音帧号）的变化是动态的，并且在统计上有如下规律：

(1) 在初始阶段（峰值点 t_0 前），其排名随着语音帧数的增加迅速后移(排名编号变大)，其后移的程度并不固定，但平均来讲不会超过词表的大小。这是因为开始时的语音还不足体现词的差异，故而正确候选的路径似然分数排名并不一定比其它的高。

(2) 在中间阶段（峰值点 t_0 以后），其排名随着语音帧号的增加开始逐渐前移（排名编号下降），而且其下降梯度很大。这是因为足够长的语音可以很好地区分不同的词，输入语音与正确候选的声学模型匹配程度越来越高。

(3) 在后面阶段，其排名随着语音帧号的增加一直稳定的排在前面，而且与之相近的几个候选路径也一直稳定的排在前列。原先排名落后很多的路径很少能在这个阶段突然排在前列。最终成为识别结果的总是在很少的前几名路径之中。

从这些实验结果中可以看出：

(1) 剪枝宽度不必固定，可以随时间减小；

(2) 要保证正确候选的路径在开始阶段留在剪枝宽度内，初始剪枝宽度要足够大，但是不必超过词表大小；

(3)排名曲线峰值点的出现帧号(t_0)（如图 3-2 所示），即初始阶段的持续时间，实验中发现与词表平均词长(Len，字数)以及输入语音总长度(T，帧数)粗略

地存在如下的对应关系：

$$t_0 \approx 1.5 \times T/(Len+2) \quad \text{(式 3-5)}$$

在已知这些情况后，我们就可以有针对性地对排名剪枝进行有效的改进。

3.3.2 时间动态调整剪枝策略

基于以上对正确候选排名趋势的实验事实出发，本文提出一种随时间动态变化剪枝宽度的直方图剪枝方法，称之为动态调整直方图剪枝。剪枝宽度在初始阶段要要足够大，其大小与词表的大小有一定的函数关系；然后剪枝的路径宽度随着输入语音帧号（时间）的增加而动态减小。该动态变化过程可视为一个非线性时变系统，为简化起见，本文利用分段线性的方法来近似描述图 3-2 所示的趋势。某个 t（单位是帧号）时刻的排名剪枝宽度 $W_d(t)$首先由如下公式计算得到。

$$W_d(t) = g(t) \quad \text{(式 3-6)}$$

其中 $g(t)$是一个输入语音帧号 t 的分段线性函数，如下定义：

$$g(t)=\begin{cases} g(t_0) & 0 \le t < t_0 \\ g(t_0)-a_1\times(t-t_0) & t_0 \le t < t_1 \\ \quad\cdots & \quad\cdots \\ g(t_{m-1})-a_m\times(t-t_{m-1}) & t_{m-1} \le t < t_m \end{cases} \quad \text{(式 3-7)}$$

这里是用 m 段线形函数来近似逼近图 3-2 中的下降趋势，显然分段线形函数的形状是由常数组$\{a_1,...,a_m\}$和$\{t_1,...,t_m\}$所决定的。通过调整这组常数可以调整剪枝宽度 $W_d(t)$下降速度，在实际中，这组常数通常是按照前松后紧的原则确定的。如图 3-3 所示的就是用一个分段线性函数来逼近的剪枝宽度随时间动态变化的示意图。

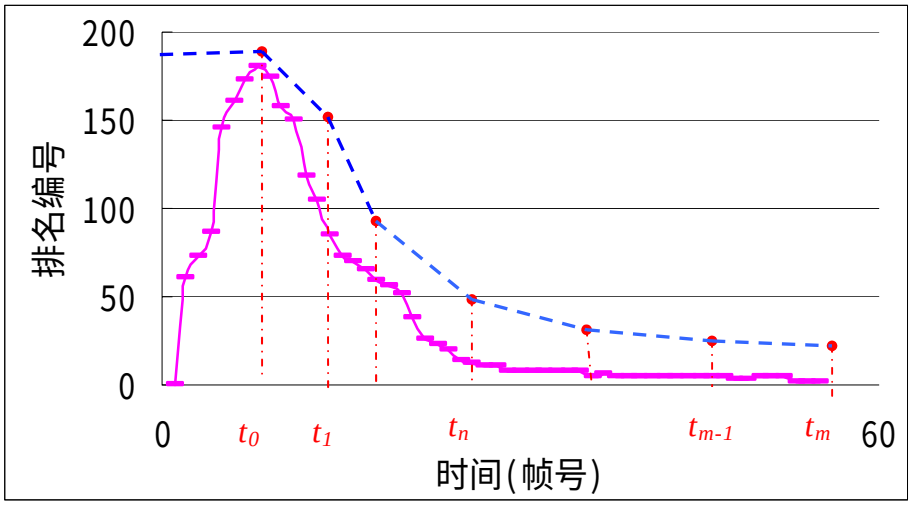

图 3-3　用分段线性函数逼近的时间动态调整剪枝

其中上方的虚线折线代表分段线性函数，而该折线上的几个圆点对应的横坐标序列代表常数组 $\{t_1,...,t_m\}$，而且每一段折线的斜率则代表了常数组 $\{a_1,...,a_m\}$，显而易见，原本需要一直维持不变的剪枝宽度，通过分段线性函数 $g(t)$对剪枝宽度的控制，减少了峰值点 t_0 后需要保留的路径个数，从而减少了路径的无效扩展和比较费时的 Viterbi 打分计算，可以达到大幅度减少解码中的打分次数的目的。

3.3.3　结合搜索路径分数差值动态调整剪枝策略

在上节中，通过分段函数的时间动态调整剪枝策略对于整个搜索的改进的是比较粗略的调整剪枝宽度。为了对剪枝宽度进行更精确的动态调整，本章继续研究剪枝过程中搜索路径似然分数的情况，并期望从中能够发现某个时刻搜索剪枝宽度与路径似然分数的关系。

实验发现，在剪枝初始阶段，因为开始时的语音比较少，还不足体现不同词之间差异，故而各个候选的路径似然分数的差别并不大，有时排名相差很多的路径之间的似然分数相差很小，但是这些排名比较落后的路径在搜索后期很可能上升为前几名，所以为了避免将其过早剪除而影响识别正确性，应该尽可能保证这些路径参与后续的路径扩展打分运算。这种情况在词表中有比较多的相同或相近发音时尤其常见。而在搜索后面阶段，随着输入语音的增多，足够长的语音已经可以很好地区分不同的词，输入语音与正确候选的声学模型匹配

程度越来越高，于是排名相差不多的路径之间的似然分数相差已经很大，并且那些排名比较落后的搜索路径的排名名次很难再上升成为识别结果。

基于上述实验事实，本文将搜索路径的似然分数情况反映到直方图剪枝的剪枝宽度调整中，期望对排名剪枝宽度进行更为精确的动态调整。

为了更精确地定义 t 时刻剪枝宽度 $W_d(t)$，本文引入 t 时刻路径的似然分数差值（即与第一名的搜索路径的分数差值）$\Delta Score(t)$：如果似然分数差值很大，说明排名比较靠后的路径成为最终结果的可能性非常小，则可以进一步加快剪枝宽度的收缩；反之，说明排名比较靠后的路径和排名靠前的路径成为最终结果的可能性非常接近，则减缓剪枝宽度的收缩。某个时刻 t 的排名剪枝宽度 $W_d(t)$ 由公式 3-8 计算得到，

$$W_d(t) = W_d(t-1) + f(\Delta Score(t)) \quad \text{(式 3-8)}$$

其中函数 $f(\cdot)$是似然分数差值 $\Delta Score(t)$的调整函数：如果该帧剪枝宽度处的路径似然分数差值大于某个分数差值阈值，则将剪枝宽度 $W_d(t)$缩小直到路径似然分数差值接近某个分数差值阈值；如果该帧剪枝宽度处的路径似然分数差值小于某个分数差值阈值，则将剪枝宽度 $W_d(t)$扩大直到路径似然分数差值接近某个分数差值阈值。这里的路径分数差值阈值可以是动态的，也可以是固定的，实验发现动态的效果会更好，即在初始阶段该分数差值阈值较小，在后续阶段较大。

综合以上两种方式的动态调整，最终在某个时刻 t 的排名剪枝宽度 $W_d(t)$可以表示为：

$$W_d(t) = W_d(t-1) + g(t) + f(\Delta Score(t)) \quad \text{(式 3-9)}$$

通过分段线性函数 $g(t)$对剪枝宽度随着时间动态变化，大幅度减少不必要的路径扩展和打分运算，同时通过分数调整函数 $f(\cdot)$对剪枝宽度在 $g(t)$调整的基础上进一步根据搜索路径似然分数差值进行比较精确的调整，既保证了搜索的快速性，也保证了搜索的正确性。

3.3.4 搜索策略比较实验

为了检验动态调整直方图剪枝方法（DHP）的性能，本文将其与标准 Viterbi

解码算法（VTB）、固定束宽度算法（FHP）进行了实验比较，其中固定束剪枝（FHP）的剪枝宽度为 50，词表大小为 200，平均词长 *Len* 为 2.7 字，设语音长度为 T 帧，则由式 3-5 知 DHP 的峰值点 $t_0 = 0.32T$，函数 $g(t)$中 $m = 4$，常数组 $\{a_1,...,a_m\}$和$\{t_1,...,t_m\}$分别为$\{100/T, 400/T, 200/T, 120/T\}$和$\{0.4T, 0.5T, 0.75T, T\}$，而相应的调整函数 $f(\cdot)$的动态阈值为$\{80.00, 90.00, 100.00, 120.00\}$，在 Dopod 686（Intel StrongARM，206MHz，32MB RAM）上得到实验结果如图 3-4 所示。其中图中括号里的数字表示搜索解码的时间，这个时间是相对的，因为对不同长度的输入语音，其搜索解码的时间可能会变化。另外图中的从左上角到右下角的斜线粗略代表识别正确率和搜索解码时间的平衡线，离平衡线垂直距离较近的说明达到识别正确率和搜索解码时间的平衡的程度较好，反之则说明较差。

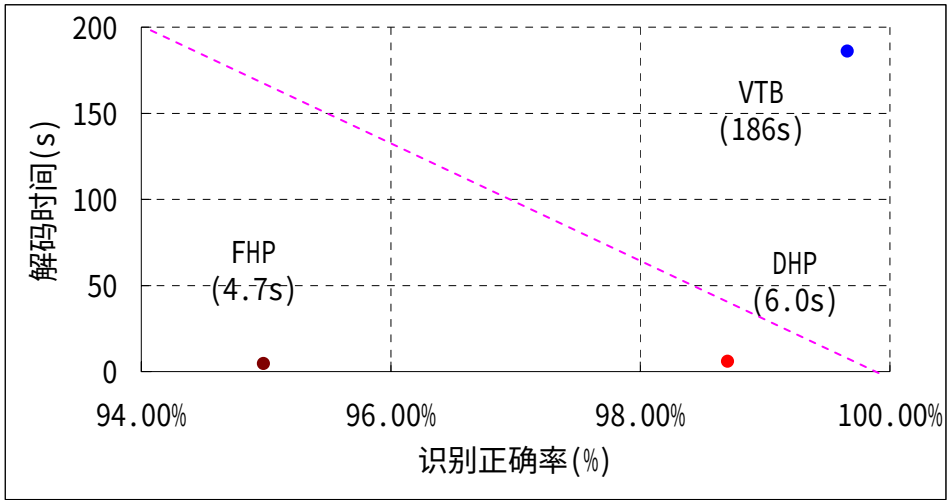

图 3-4　不同搜索策略的性能比较

显见，在剪枝解码过程中，影响识别正确率的最重要的因素是在初始阶段设置合适的剪枝宽度以便把正确候选包含进去；而影响速度的最重要因素是在后续阶段将剪枝宽度有效地降下来。从实验结果来看，FHP 方法虽然在节省运算上很有效，但是因为剪枝宽度固定不变而且在初始阶段太小，导致本来能成为正确结果的路径过早被剪枝掉；当然，如果本文的实验选取固定剪枝宽度大，则可能的结果是正确率提高但解码速度降低。

3.4 词典表示

在具体实现某种搜索算法时，我们还要考虑的就是词典的表示问题，它关系到整个搜索空间的规划从而显著影响搜索算法的效率。通常的词典表示可以采用线性词典(Linear Lexicon)和树型词典(Tree Lexicon)[40][41][42][43]两种方式，见图 3-5 和图 3-6 所示。

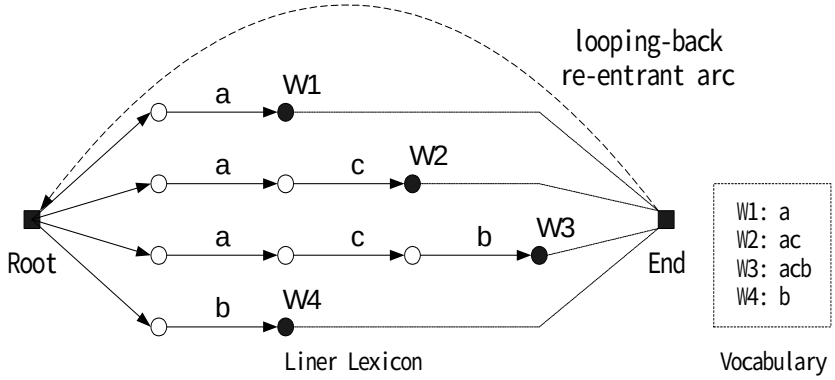

图 3-5 线性词典的例子

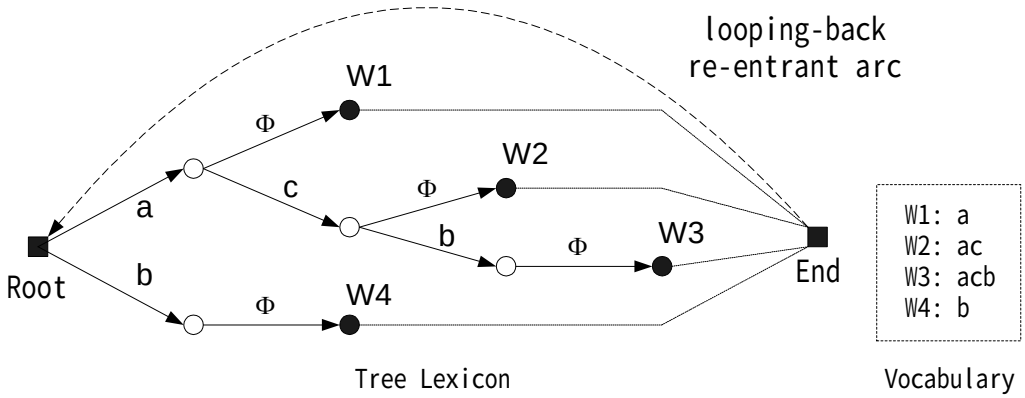

图 3-6 树形词典的例子

对于线性词典，各个词的模型在搜索的过程中保持严格的分离，词的模型之间没有信息的共享。而对于树型词典，由于它对线性词典中具有相同前缀的词进行了合并，因此可以在一定程度上降低词典表示的存储空间，同时降低后续 Viterbi 解码过程中搜索空间的规模。在嵌入式系统上进行语音命令识别的时候，为了节省搜索解码过程中的存储和计算，采用比较经济高效的树形词典来

进行搜索是非常必要的。为此我们在模拟器上进行了对比实验验证，实验结果如图 3-7 所示。

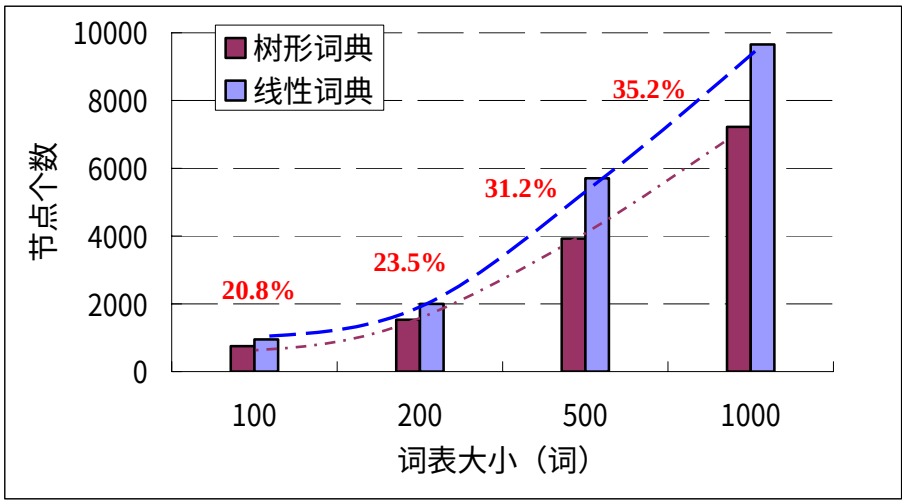

图 3-7　不同词典表示方式的存储空间比较

从我们在模拟器上的实验可以发现：一般情况下，对于相同的词表情况，随着词表大小的增加，树形词典的表示方式比线性词典节省存储空间的比率按照接近线性的增加；同时如果后续搜索方法采用束搜索剪枝策略，那么词典节点数目的减少对于束搜索的性能将有很大的提高，尤其是在词典规模越来越大的时候，效果更为明显。

3.5　打分速查表

为了进一步提高搜索的效率,减少搜索空间的存储，本文除了采用树型词典表示外，还提出对搜索中的大量状态似然打分计算利用速查表进行优化。

实验研究发现，在搜索解码过程中，计算每帧输入语音与所有候选语音模型每个状态的匹配得分过程包含了大量的重复计算（相同模型节点或状态节点的）。而一般而言，在搜索过程中，状态似然分数的计算量占整个搜索过程计算量的 80%以上[44]。针对这种情况，本文提出采用数据共享技术的思路，以表格形式（即速查表）动态存储词表中节点模型的状态与每帧输入语音的匹配得分，在计算每帧输入语音与所有候选语音模型某个状态的匹配得分时，如果该状态在速查表中已经存在（即前面已有打分过），则直接得到似然得分；否则，

实际计算该状态的似然得分，并更新记录在速查表中。通过这种速查表的方法，可以节省搜索中的大量重复似然分数计算，从而也大大地提高了搜索解码的速度。

在采用动态调整直方图剪枝的搜索策略的前提下，本文对是否采用速查表共享技术在不同词表大小的情况进行了对比实验，实验结果如图 3-8 所示，其中的百分数表示采用速查表共享技术后节省计算量的程度。

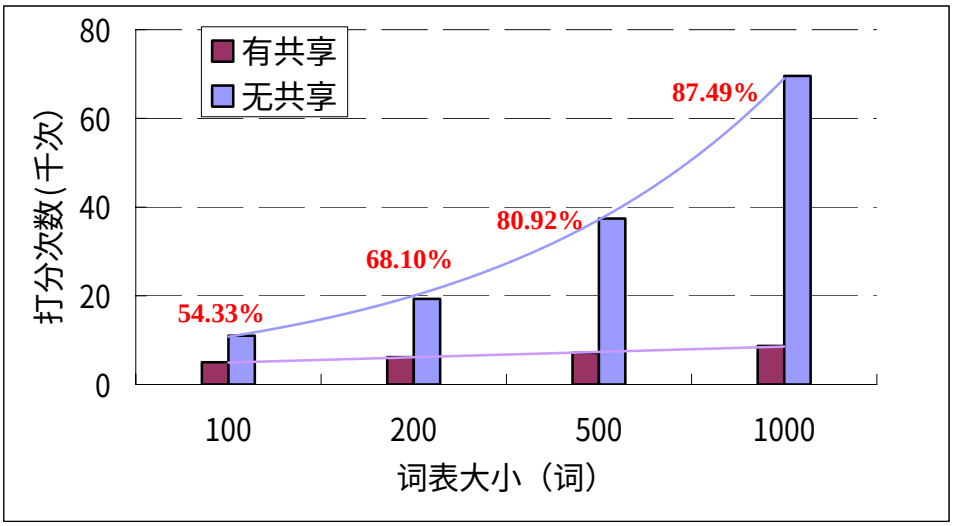

图 3-8　数据共享技术对搜索的优化实验

实验表明，不采用速查表共享技术时，打分次数随着词表大小以二次多项式增大；而采用速查表动态共享技术后，当词表较小时，打分次数以低斜率增长，但当词表大小继续增大后，则保持在一个比较稳定的范围，因为当词表涵盖了所有的模型状态后，搜索解码过程中速查表动态共享技术能够保证每个模型状态对每一帧输入语音只计算一次，那么总的打分次数就只和模型状态总数和输入语音长度有关。

3.6 本章小结

本章针对嵌入式语音命令识别系统中的搜索解码的速度和存储空间问题，主要通过如下研究进行了改进。

(1) 研究了正确候选搜索路径在 Viterbi 束搜索中的排名顺序与输入语音帧数的动态变化关系，提出了随时间动态变化剪枝宽度的动态直方图剪枝策略；研究了 Viterbi 束搜索中排序后相邻排名的搜索路径分数差值随时间变化的关系，进而提出了一种结合搜索路径分数差值动态调整直方图剪枝的搜索策略，减少了不必要的搜索路径保留，提高了解码的速度，同时保证了剪枝的正确性；

(2) 研究了搜索解码过程中的似然分重复计算问题，提出在 Viterbi 解码中利用速查表加速计算的方法，减少了冗余的似然分数计算，提高了解码的速度；

(3) 以及其他利用树形词典，模型级剪枝等方法来减少搜索解码中的存储；

经过实验比较验证，证明了本章针对嵌入式提出的搜索解码策略的有效性和实用性。

第4章　嵌入式系统上的端点检测改进

4.1　端点检测在语音识别中的作用

在语音识别系统中，数字语音信号是由语音、静音和各种背景噪音等混合组成的。在这样的信号中，将语音和各种非语音信号时段区分开来，准确地确定出语音信号的起始点称之为端点检测[45]。

从图 4-1 的语音识别系统的各个功能组成部分可以看出，端点检测在整个语音识别系统中是第一个进行处理语音信号输入的部分，对于后续的特征提取以及搜索解码有很大的影响。

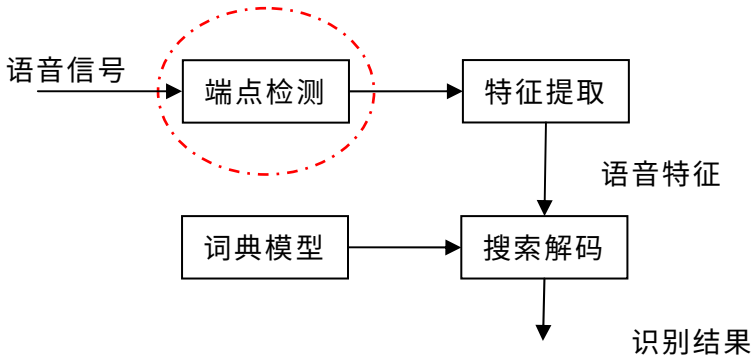

图 4-1　端点检测在语音识别系统中的地位

在语音识别中，端点检测的性能对于识别的正确率，识别速度都有很大的影响[46]。表现在以下几个方面：

(1) 在语音识别和说话人识别中为了消除信道的影响通常采用倒谱均值相减的方法，需要对语音时段的端点准确定位。

(2) 为使整句的似然得分累计更多的集中在语音段，不被噪音所分散，必须去除掉静音段，这样有助于识别率的提高。

(3) 在不断变换的环境下对噪音和静音建模是非常困难的。准确的端点检测能够事先移除单纯噪音的时段，这对于噪音和静音模型的精确建立有很大帮助。

(4) 当所处理信号含过长的非语音时段时，准确的端点检测可以极大提高计算速度。

(5) 对于某些语音识别系统，比如开放式语音识别系统、自适应语音增强系统、语音信号传输系统等，端点检测的准确与否对于系统性能有着重要影响。在开放式语音系统中，自适应增强算法需要准确的标出噪音段用作噪音谱的自适应估计，在语音信号传输中，例如开放式广播语音信息的传输，好的端点检测能极大的降低所要传输的信息量。

4.2 端点检测的基本方法分类

到目前为止，端点检测的些方法大致可以分为两大类：基于特征的方法和基于模型的方法。

基于特征的方法又可分为基于鲁棒特征的方法和特征滤波的方法。基于鲁棒特征思想的出发点是寻找能表征语音和噪音在不同域差异的特征来进行语音和噪音时段的区分，所用的特征主要有能量[47]，子带能量[48]，过零率[49] ，基频[50]，周期度量，熵[51] [52]，能量方差等。基于特征滤波的思想的出发点是对特征先进行滤波，然后进行端点检测，主要算法有子空间滤波[53] ，能量差分自适应滤波[54]等。这些方法的缺点分别是：能量方法在高信噪比条件下效果很好，随着噪音环境恶化性能急剧下降；基于子带能量，过零率，周期度量，基频的方法只适用于某些种类的噪音环境；熵的方法在 babble 噪音环境下效果不好；基于特征滤波的方法一方面增大了计算量，另一方面改变了语音谱的结构，丢失了部分信息。

基于模型思想的出发点是针对噪音和语音进行建模，用来区分语音时段。基于模型的方法的缺点在于噪音的环境多种多样，不可能对各种情况都建立相应的模型。当噪音环境与模型不匹配时，性能严重退化。

4.3 嵌入式声控拨号系统中的端点检测

在实际嵌入式声控拨号系统应用中，语音识别引擎的载体 PDA 是漫游的，所处的噪音环境是随机且不可预测的，噪音的种类、信噪比都有可能急剧变化。这就要求端点检测算法具有以下性能：对不同噪音不同信噪比的鲁棒性，较低的计算复杂度，快速的响应性能，精确的检测性能，简单的实现。

从上节各种端点检测的方法的优劣介绍和比较，本文针对嵌入式系统运算速度比较慢的特点，选择了 Lawrence Rabiner 提出的经典端点检测方法——结合语音短时能量和过零率作为语音和噪音判断的依据，建立完善的实时端点检测算法，并针对具体应用中出现的问题提出了几点改进，取得了比较好的效果。

4.3.1 结合短时能量和过零率的端点检测

结合短时能量和过零率的端点检测的方法是根据语音信号短时平稳的特性，利用语音声韵母的能量特性和过零率特性，计算每一段短时信号的能量及其过零率后，将其与既定的阈值比较，超过阈值则判定为语音，否则判定为噪声。

其中，短时能量（Energy, E）的定义为：在统计的短时段的信号幅度平方和。过零率（Zero-Crossing-Rate, ZCR）的定义为：在统计的短时段中，信号波形穿越零电平的次数。

记 x(n)为离散语音信号时间序列，假设长度为 N(1,2,…,N)，则短时信号的能量 E 和过零率 ZCR 的计算公式分别为：

$$E = \sum_{m=-\infty}^{+\infty} [x(m)]^2 \qquad \text{(式 4-1)}$$

$$ZCR = \sum_{m=1}^{N} |\mathrm{sign}[x(m)] - \mathrm{sign}[x(m-1)]| \qquad \text{(式 4-2)}$$

其中，函数 sign(*x*)为符号函数。

4.3.2 改进的端点检测方法

针对实际的嵌入式应用，本文采取了一些改进措施来提高端点检测的速度和效果。

(1) 消除零漂移和简化计算。在计算信号的短时能量的时候，由于要计算幅度的平方和，不仅计算量比较大，而且由于信号有零漂移的影响，这样得到的信号能量用来作为端点检测的判决标准并不准确。在实际应用中，为了提高端点检测的计算速度，简化了能量的计算方式，本文采用了信号与平均幅度值的差值 AVG(*E*)的和来代替信号幅度的平方和作为短时信号的能量，平均幅度值的差值 AVG(*E*)及改进公式为：

$$AVG(E) = \frac{1}{N}\sum_{m=1}^{N} x(m) \quad \text{(式 4-3)}$$

$$E = \frac{1}{N}\sum_{m=1}^{N} |x(m) - AVG(E)| \quad \text{(式 4-4)}$$

采用了以上计算能量的做法后，可以有效的消除端点检测过程中的零漂移现象，并提高了计算的速度。

(2) 对端点检测判决阈值的自适应更新。我们知道，在对短时信号进行判断是语音或是噪声的时候，需要用能量和过零率对判决阈值进行比较，所以这个阈值对于端点检测的性能具有很大的影响。本文采用的端点检测阈值是一个自适应更新的端点检测阈值：首先利用输入语音的前几帧，得到一个初始的能量阈值和过零率阈值，作为接下来的判决阈值；在得到后续的信号段的类别后，利用这些信号中非语音段的能量和过零率，更新现在的端点检测阈值，进行后续的判断。改进后的整个端点检测的流程以及某一帧的 VAD(Voice Activity Detection)判决如图 4-2 和图 4-3 所示。

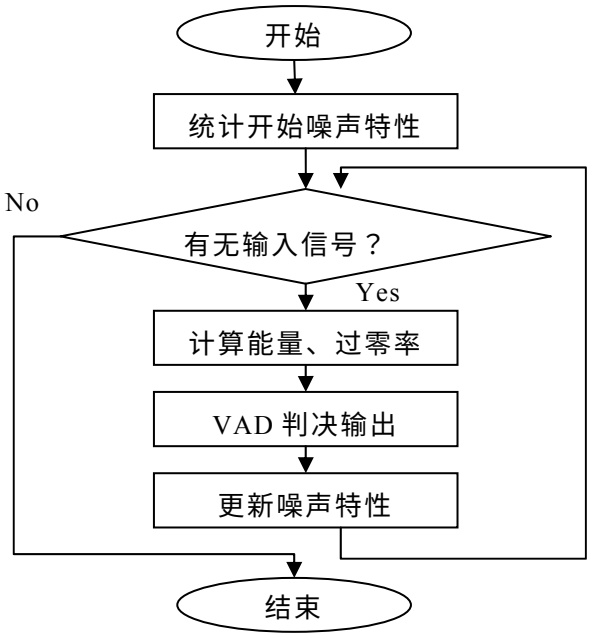

图 4-2　端点检测算法流程

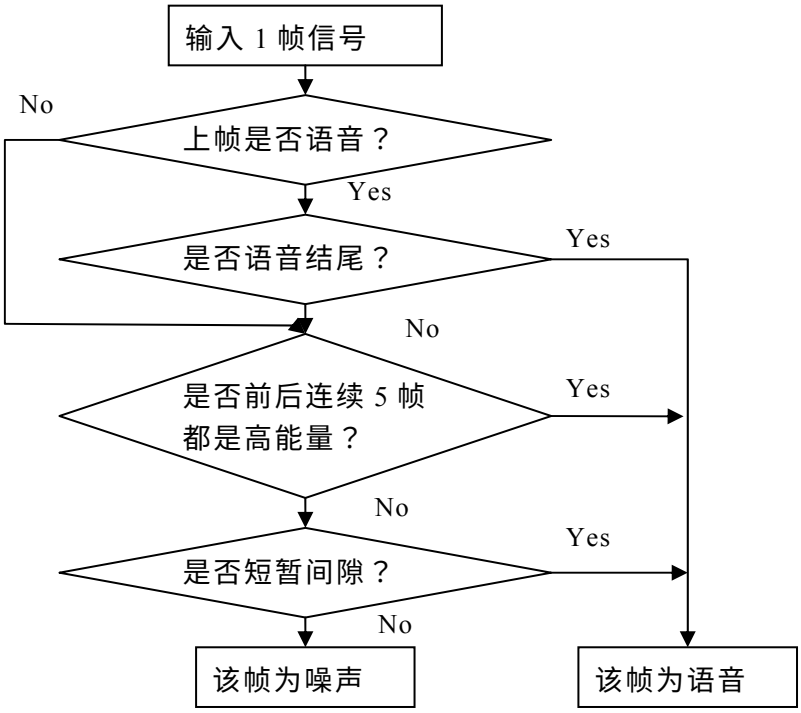

图 4-3　VAD 判决输出算法流程

（3）克服开始阶段的纯零幅度信号。在实际使用中，我们发现起始阶段的信号的幅度很可能为纯零，那么在计算该段信号的过零率和能量的时候，就有可能为 0 或十分接近 0 的数值，这样用来产生初始阶段的端点检测的阈值就会变得很小，此时如果后续噪音帧中过零率略有增大，则很容易使系统过度敏感从而导致起点识别不准。针对这种信号纯零幅度的情况，本文在更新端点检测阈值的时候，增加了对最小值的判断，如果属于这种纯零的情况，则不能用该段得到的很小的能量和过零率阈值作为新的判决阈值。

4.4 本章小结

本章主要研究了嵌入式声控拨号系统中的端点检测部分，针对实际嵌入式设备的应用环境特点，选择了运算速度比较快的端点检测算法，并针对运算速度的问题，采用了改进的短时能量计算方法；同时，针对实际使用过程中出现的零漂移、开始阶段的纯零幅度信号等情况，采取了相应的改进措施，取得了比较好的效果。

第5章　声控拨号系统设计及性能评测

随着移动通信设备的迅猛发展，通信技术的更新换代，以及语音识别技术的发展，现在的移动设备，比如高端手机、PDA 等，都开始兴起增加声控拨号的功能，方便用户从纷繁众多的电话本联系人中挑出想要呼叫的联系人。本章将介绍本文设计并实现的嵌入式声控拨号系统---“ 得意声控联系人 ”(d-Ear Voice Contacts)，适用于个人数字助理 PDA。

5.1　声控拨号系统功能概述

5.1.1　声控拨号概述

声控拨号（也叫语音拨号）的定义就是指用户摘机后说出被叫者姓名，电话设备即自动拨出被呼叫者的电话。比如：“ 张三 ”的电话是“ 8612345678910 ”，只要用户摘机后说：“ 张三 ”，电话设备将自动拨通“ 张三 ”的电话“ 8612345678910 ”，无需挨个去输入电话号码或者从众多的电话本联系人中寻找出“ 张三 ”这个人。移动设备上带有声控拨号功能可以大大的节约用户的时间和精力。

在早期，具有声控拨号功能的手机，要求用户在使用声控拨号之前，必须预先录制声控标签，也就是说为电话簿中的几个电话号码录制对应的语音作为标签存入话机，在声控拨号的语音识别过程中，系统将对用户输入的语音和原有的声控标签语音进行模板匹配进而得到识别结果，这种方法大部分的识别效果都不尽理想，而且随着声控标签的增多，所占存储空间变得越来越大，严重限制了可呼叫的联系人的数目。

5.1.2　“ 得意声控联系人 ”功能简述

本文实现的声控拨号系统---“ 得意声控联系人 ”是基于 Pocket PC 操作系统的嵌入式软件，主要用于个人数字助理 PDA 上的汉语语音拨号。它随 Pocket PC 的“ 联系人 ”打开而自动加载；当用户需要查找某特定联系人时，可以通过按键方式直接呼叫要查找的联系人姓名，声控程序将会自动找到目标联系人。图

5-1 展示的是一次声控拨号的情况。

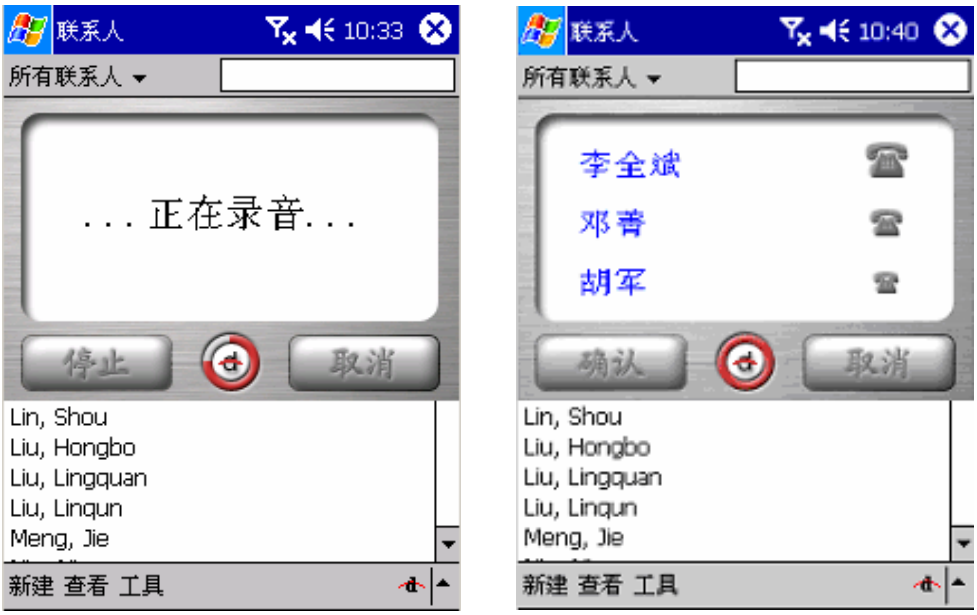

图 5-1 “得意声控联系人”使用图示

“得意声控联系人”的姓名列表来自 Pocket PC 的联系人姓名(的部分或全部)，在构建姓名列表时允许对联系人姓名中的多音字进行读音选择；而且其姓名列表可以进行修改，包括自定义姓名的读音，如用昵称来代替姓名原来的读音等。其次，它是与说话人无关(即非特定人)的语音识别系统，无需在线训练，无需预先录制声控标签。

总的来讲，“得意声控联系人”具有如下特性：

(1) 完全嵌入式界面，自动加载运行；

(2) 非特定人，无需训练；

(3) 联系人姓名的读音可定制；

(4) 识别率高，识别速度快，模型小(模型 290 KB，程序 820 KB)；

5.2 声控拨号系统架构及设计

本节将介绍声控拨号系统---“得意声控联系人”的架构及设计过程。

5.2.1 系统功能分解

首先，我们来看一下声控拨号功能的主要处理流程：

(1) 启动语音拨号，直接说出要呼叫的电话本中某个联系人的名字；
(2) 系统对用户的语音信号进行端点检测，去除首尾非语音的部分；
(3) 对语音部分进行特征提取，得到特征向量；
(4) 将得到的特征向量送入识别器，进行搜索解码，得到识别结果；
(5) 快速得到该联系人对应的电话号码，进行选择呼出拨号。
系统主要处理流程如图 5-2 示。

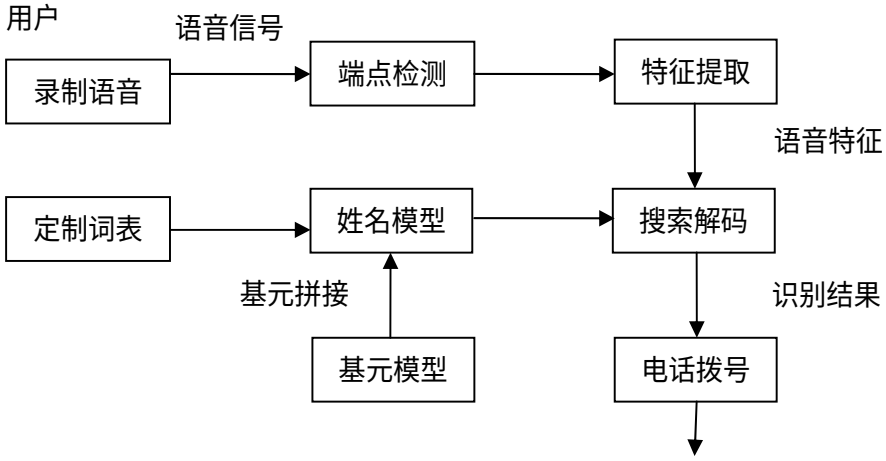

图 5-2　声控拨号系统处理流程

根据以上对于声控拨号系统主要处理流程的分析，系统功能可以分解为四大主要部分：录制输入语音、识别输入语音、实际拨号、以及定制联系人词表等。在这四个功能中，识别输入语音部分主要涉及到语音识别核心引擎，功能最复杂，而且与界面处理无关，而其他几个部分都和界面以及实际设备有关，功能相对简单，当然实际实现过程中发现界面的处理也是一件很复杂的事情。本节的接下来部分将主要介绍语音识别核心引擎部分的架构和设计过程。

5.2.2　语音识别核心引擎架构设计

在语音命令识别系统中，语音识别核心引擎的功能基本上可以分为以下几个部分：端点检测、特征提取、生成词典、搜索解码。这四个部分之间的关系如图 5-3 所示，是相互关联的，有着一定的先后关系和依赖关系。

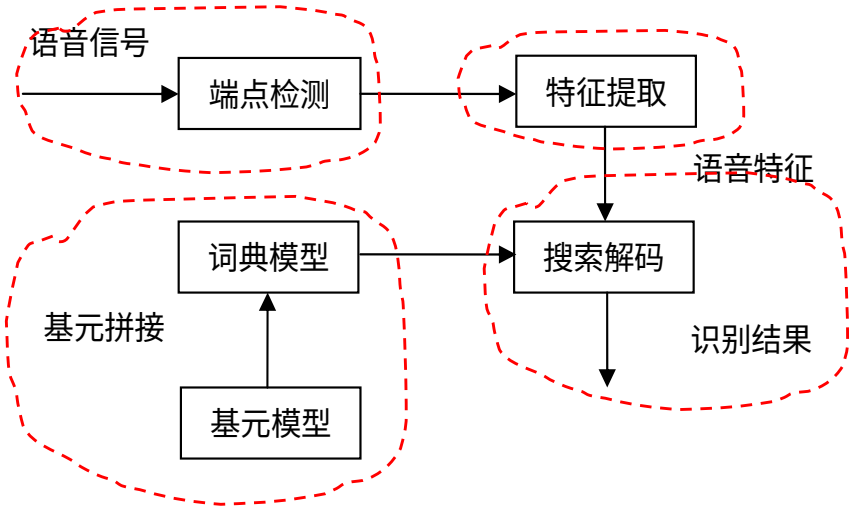

图 5-3　语音识别核心引擎功能流程分解

根据以上几个功能分解，本文设计实现的“得意声控联系人”系统的语音识别核心引擎架构可以因此分解为如图 5-4 所示几个模块形式：搜索解码模块、特征提取模块、生成词典模块、模型管理模块、端点检测模块。图中的箭头表示模块之间的依赖和调用关系。

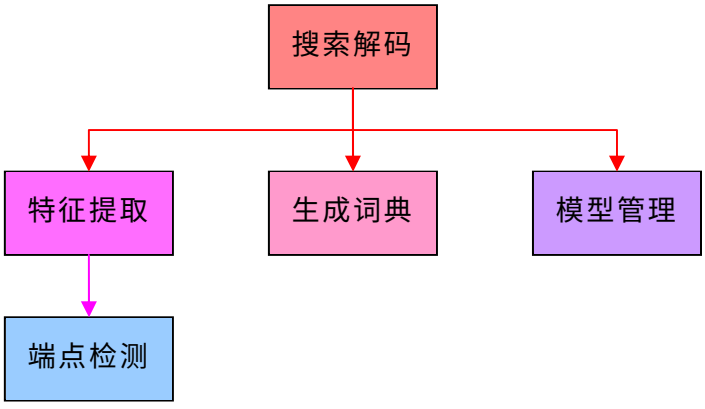

图 5-4　语音识别核心引擎架构分解

各个模块的功能描述如下：

(1) 端点检测模块：根据输入的语音数据，检测出首尾的静音长度，并将得

到的结果返回给特征提取模块。

(2) 特征提取模块：根据输入的语音及其首尾静音的长度，对语音部分进行根模型特征相匹配的特征提取，得到特征序列，输出给搜索解码模块进行识别。

(3) 生成词典模块：根据输入的用户词条（汉字或拼音词条），将这些词条分解为和基元模型一致的基元序列拼接，比如得到这些词条的声韵母序列，并将这些结果返回给搜索解码模块用于构建搜索网络。

(4) 模型管理模块：根据 HTK 的模型形式，将已经训练好的基元模型载入，并组织存储各个基元模型的参数，并提供计算输出概率、转移概率的功能，最终将这些结果返回提供给搜索解码模块。

(5) 搜索解码模块：根据输入的语音特征向量序列和用户词典、模型参数构建搜索识别网络，并进行最后的搜索剪枝、得到语音识别结果返回给应用上层。

5.3 声控拨号系统性能评测

本节将介绍声控拨号系统的性能测试情况。

本文设计实现的“得意声控联系人”的声学模型训练集都从电话信道的数据库中得到。具体的训练数据情况如下：

(1) 所有的语音都是在较低噪音的环境下录制而成，采样率为 8kHz，采样深度为 16 位。

(2) 训练数据库包括 15 人 863 男声数据，15 人 863 女声数据，150 人电话数据，总共约 30,000 多句。

(3) 训练数据的语料内容基本为短句子、打电话说“我找某某某”之类的语句，以及单个孤立词等等，大部分属于连续语音的形式。

(4) 说话的方式分布情况为：863 数据库为较为清晰，语速较慢的朗读形式；打电话找人的数据库为比较随意，有些发音不是很清晰，语速较快的自然说话方式。

(5) 语音特征情况为：采用 MFCC 特征，为“{9 维 MFCC+1 维能量}+一阶差分”一共 20 维特征。帧长为 24ms，帧移为 12ms。

(6) 模型的训练方式为采用 HTK 工具进行训练。

在从上述的训练数据得到训练好的声学模型后，在声控拨号系统中加载使用，本文准备了一些测试数据对系统的性能进行测试。这些测试数据的情况为：

语音数据在 Dopod 掌上电脑（采用 Pocket PC 2002 操作系统和 StrongARM 芯片）上通过 8K 采样率，16 位采样深度，实地录制得到，由 6 个人（4 男 2 女）每人说一遍词表中的人名，一共 1,175 句，每个声音录制长度约为 3 秒。词表为 200 个随机选取的中文人名，主要是 150 个三字词和 50 个两字词。录音环境为有点噪音的办公室环境，主要噪音为电脑的风扇转动声音、敲打键盘的声音、有时有电话铃声、有些周围有其他人的说话声等。

测试机器为多普达 PDA 掌上电脑，采用 StrongARM 芯片，处理器频率为 400MHz，内存为 SDRAM128MB，操作系统为基于 Windows CE 的 Pocket PC 2003 系统。

对以上 1,175 个测试数据进行词表大小为 200 词的性能测试结果如下表 5-1 所示，其中 d-Ear VC 是本文实现的得意声控联系人的测试结果，Baseline 是本文采用标准 Viterbi 解码算法的测试结果。

表 5-1　声控拨号系统性能测试

	搜索时间/s	识别正确率/%	搜索空间/KB
d-Ear VC	2.3	98.72	120
Baseline	187.0	99.66	190

从以上对大量测试数据的系统测试结果表明：本文设计并实现的声控拨号系统在中小词汇量的情况下，具有较高的识别正确率，以及较好的识别速度，相对于基于标准算法的系统，识别速度可提高约 80 倍，而搜索存储空间可节省约 30%。

5.4 本章小结

本章介绍了本文设计并实现的嵌入式语音命令识别系统的一个实际应用成果---“得意声控联系人”，首先介绍了系统的总体功能，然后介绍了系统的功能分解以及架构设计，重点介绍了语音识别核心引擎的设计和实现，最后是系统的实际性能测试评价，当词表随机选取为 200 个中文人名时，识别正确率可达到 98.70%，相对于基于标准算法的系统，识别速度可提高约 80 倍，而搜索存储空间可节省约 30%。。

第6章　结论

6.1　工作总结

本文以嵌入式操作系统系统 Pocket PC 为实验平台研究基于非特定人语音命令识别的声控拨号系统（声控拨号器）。针对在 PDA 上的存储空间和运算速度的限制条件，在不影响识别率的基础上，降低声控拨号系统的时空复杂度。

（1）研究了在不特别降低识别性能的情况下如何选用合适的声学识别基元，以及采用较少的特征维数进行声学建模。通过实验比较选定扩展声韵母作为识别基元，并对传统 PC 上使用的多维特征进行压缩，提出一种把基于基元拼接、整词识别的声学建模技术用于嵌入式声控拨号系统的方案。

（2）研究了正确候选搜索路径在 Viterbi 束搜索中的排名顺序与输入语音帧数的动态变化关系，同时研究了排序后相邻排名的搜索路径分数差值随时间动态变化的关系，进而提出了一种结合搜索路径分数差值进行调整的动态调整直方图剪枝的搜索策略，减少了搜索解码中的冗余计算量，提高了搜索解码的速度。

（3）研究了搜索解码过程中的似然分重复计算问题，提出在 Viterbi 解码中利用速查表加速计算的方法，进一步提高搜索解码的速度。

（4）研究了传统端点检测技术在实际嵌入式应用中出现的零漂移、开头纯零等问题，对端点检测算法提出了改进方法来提高端点检测的准确性。

经过实验研究，本文综合应用上述方法以及其他实现技巧，设计并实现了一个非特定人的可定制词表的语音命令识别系统—“得意声控联系人”，改进了系统的性能，并将其实用化和推广。

当词表随机选取为 200 个中文人名时，识别正确率可达到 98.70%，相对于基于标准算法的系统，识别速度可提高约 80 倍，而搜索存储空间可节省约 30%。

6.2 需进一步开展的工作

对于未来工作的展望，除了使声控拨号在嵌入式环境下的更加快速和节省资源，还需要从以下方面进行改进：

1）克服环境噪声的影响。PDA 作为一种移动通讯工具，任何的使用环境都有可能出现，尤其是噪声环境，需要增强系统的抗噪性能。

2）克服用户口音的影响。让更多的用户能够随心所欲地使用声控拨号。

3）可以进一步从定点运算上进行搜索速度的提高。

随着移动设备的技术迅速发展，语音识别的技术难点也正在不断突破中，相信以上的困难也将会逐步被解决。

参考文献

[1] Brian Kronstad，嵌入式：后 PC 时代的技术主力，http://www.people.com.cn/GB/it/1065/2327077.html

[2] 朱璇 刘加 刘润生，语音识别技术新热点 — 语音识别专用芯片，世界电子元器件

[3] B.H. Juang, The past, present and future of speech processing, IEEE Signal Processing Magazine, May, 1998；

[4] 陈德锋，郑方，吴文虎等，动态调整直方图剪枝 PDA 声控拨号器的应用与实现，电声技术

[5] 郑方，胡起秀，邓翔等，介绍一种傻瓜式声控电话机，第四届全国人机语音通讯会议，1996 年 10 月，北京

[6] 朱纯益，陆建华，刘润生等，基于 DSP 的声控电子记事本的设计与实现，电子技术应用，2002 年第 9 期

[7] 荆嘉敏，刘加，刘润生等，基于 HMM 的语音识别技术在嵌入式系统中的应用，电子技术应用，2003 Vol.29 No.10

[8] 邓勇刚，徐 波，黄泰翼，Palm PC 语音识别算法及实现，计算机研究与发展，2000 年 8 月

[9] 谢凌云，杜利民，刘斌，嵌入式语音识别系统的快速高斯计算实现，计算机工程与应用，2004 年第 23 期

[10] 张继勇，汉语语音识别中声学建模及参数共享策略的研究（硕士学位论文），清华大学计算机科学与技术系，2001

[11] Viterbi, A.J., Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, IEEE Trans. on IT, 13(2), 1967

[12] Lee, C.H., Rabiner, L.R., A frame synchronous network search algorithm for connected word recognition, Proc. of IEEE, 77(2), pp.257-285, 1989

[13] 方敏，浦剑涛，李成荣等，嵌入式语音识别系统的研究和实现，第七届全国人机语音通讯会议，2003，厦门

[14] Huang, X.D., Jack, M.A., Semi-continuous hidden Markov models for speech signals, Computer Speech and Language, vol.3, pp.239-251, 1989

[15] Zheng, F., Mou, X.L., Wu, W.H., et al, “On the embedded multiple-model scoring scheme for speech recognition”, International Symposium on Chinese Spoken Language Processing (ISCSLP’98), ASR-A3, pp49-53, Singapore, 1998

[16] 牟晓隆，汉语语音听写机的研究与实现，（硕士学位论文），清华大学计算机科学与技术系，1998

[17] Fang ZHENG, Wenhu WU, Ditang FANG. A New Model for Speech Recognition: Center-Distance Continuous Probability Model. CJSLP’97, 187-192, Mar. 1997, Huangshan

[18] 郑方，吴文虎，方棣棠，汉语语音听写机中的语音识别基元，第四届全国人机语音通讯学术会议（NCMMSC-96）论文集，pp.32~35, 1996，北京

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

[20] Li, J., Zheng, F., and Wu, W.H., “Context-Independent Chinese Initial-Final Acoustic Modeling”, International Symposium on Chinese Spoken Language Processing (ISCSLP’00), pp. 23-26, Beijing, 2000

[21] Zhang G.L., Zheng F., Wu W.H., A Two-Layer Lexical Tree Based Beam Search in Continuous Chinese Speech Recognition. In: Proc. EuroSpeech2001, Aalborg, 2001. 1801-1804

[22] 张国亮，郑方，基于两层词法树的大词表连续语音识别搜索算法，第六届全国人机语音通讯学术会议，2001 年，深圳

[23] 李峰，浦剑涛，李成荣等，基于声韵母建模基元拼接和整词识别的非特定人孤立词语音识别系统的研究，第七届全国人机语音通讯会议，2003，厦门

[24] Zheng, F., Song, Z.-J., and Xu, M.-X., EASYTALK: A large-vocabulary speaker-independent Chinese dictation machine, EuroSpeech‘99, Vol.2, pp.819-822, Budapest, Hungary, 1999

[25] Yong, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V. and Woodland, P., “The HTK Book (for HTK Version 2.2)”, Cambridge University (1999)

[26] Davis, S.B., Mermelstein, P., Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences, IEEE Trans. on ASSP, Aug. 1980

[27] 张贤达，现代信号处理，北京：清华大学出版社，1995 年

[28] Fang Zheng, Guoliang Zhang. Integrating the energy information into MFCC, International Conference on Spoken Language Processing (ICSLP’00), pp. I-389~292, Oct. 16-20, Beijing

[29] 张国亮，大词表连续语音识别中搜索算法的研究与实现（硕士学位论文），清华大学计算机科学与技术系，2001

[30] Paul D.. An Efficient A* Stack Decoder Algorithm for Continuous Speech Recognition with A Stochastic Language Model. In: Proc: ICASSP1992, San Francisco, CA, 1992. 25-28

[31] Kenny P.,. A* - Admissible Heuristics for Rapid Lexical Access. IEEE Trans. On Speech and Audio Processing, 1993. v1

[32] Lee, C.H., Rabiner, L.R., A Frame Synchronous network search algorithm for connected word recognition. IEEE Trans. Acoustic Speech and Signal Processing. 1989. v37(11): 1649-1658

[33] Jeliner F., Continuous Speech Recognition by Statistical Methods. In: Proc. IEEE, vol. 64, pp.532-556, Apr.1976

[34] Lee, C.H., Rabiner, L.R., A Frame Synchronous network search algorithm for connected word recognition. IEEE Trans. Acoustic Speech and Signal Processing. 1989. v37(11): 1649-1658

[35] Imre Kiss, Marcel Vasilache, Low Complexity Technique for Embedded ASR System, ICSLP, Denver, Colorado USA, 2002

[36] Wilpon, J.G., Lee, C.H., Rabiner, L.R., (1989) "Application of Hidden Markov Models for Recognition of a Limited Set of Words in Unconstrained Speech," ICASSP-89, 3: 254-257

[37] Suontausta J., Fast decoding in large vocabulary name dialing, ICASSP, 2000

[38] 张东滨，杜利民，语音识别的自适应束剪枝方法，电声技术，2004 年第 8 期

[39] Janne Suontausta, Fast Decoding Techniques for Practical Real-time Speech Recognition System, IEEE Workshop ASRU99，Keystone，Clorado, 1999

[40] Ney H., A Word Graph Algorithm for Large Vocabulary, Continuous Speech Recognition. In: Proc. ICSLP1994, Yokohama. 1994. 1355-1358

[41] Haeb-Umbach R., Ney H., Improvements in Time-synchronous Beam Search for 10000-word Continuous Speech Recognition. IEEE Trans. Speech Audio Processing. 1994. v2, 353-356

[42] Ortmanns S., Ney H., Experimental Analysis of the Search Space for 20000-word Speech Recognition. In: Proc. EuroSpeech1995, Madrid, Spain, 1995. 901-904

[43] Mitchell, C.D. & Setlur, A.R., Improved Spelling Recognition Using A Tree-based Fast Lexical Match. In: Proc. ICAASP1997

[44] Imre Kiss, Marcel Vasilache, Low Complexity Technique for Embedded ASR System, ICSLP, Denver, Colorado USA, 2002

[45] X. Huang, A. Acero, H. Hon，Spoken Language Processing: A Guide to Theory, Algorithm, and System Development，Prentice Hall, 2001

[46] 王卓，李鹏，苏牧，徐波，噪音环境下基于高阶谱的端点检测算法，第七届全国人机语音通讯会议，2003，厦门

[47] J. G. Wilpon, L. R. Rabiner, and T. Martin, An improved word detection algorithm for telephone-quality speech incorporating both syntactic and semantic constraints, AT&T Bell Lab Tech. J, vol 63 pp 479_498, Mar, 1984

[48] J. A. Haigh and J. S. Mason, Robust voice activity detection using cepstral features Proc IEEE TENCON, 1993, pp 321-324

[49] J. C. Junqua, B. Reaves, and B. Mak, A study of endpoint detection algorithms in adverse conditions: Incidence on a DTW and HMM recognize" Proc, Eurospeech, 1991, pp. 1371-1374

[50] R. Chengalvarayan, Robust energy normalization using speech/nonspeech discriminator for German connected digit recognition", Proc Eurospeech99 pp 61-64

[51] Jialin Shen, Jeihweih Hung, Linshan Lee, Robust entropy_based endpoint detection for speech recognition in noisy environments", International Conference on spoken Language Processing, Sydney 1998

[52] Chuan Jia, Bo Xu, An improved entropy based end point detection algorithm, ISCSLP 2002

[53] D. Wu, R. Chen A Robust Speech Detection Algorithm for Speech Activated Hands-Free Applications, ICASSP 99, vol 4 pp 2283-2286

[54] Qi Li, Jinsong Zhen, Qiru Zhou Robust Endpoint Detection and Energy Normalization for Real-Time Speech and Speaker Recognition, IEEE Tran on Speech and Audio Processing, Vol 10 No 3, March 2002

致　谢

衷心感谢导师吴文虎教授和实验室主任郑方教授对本人的精心指导。他们渊博的知识、严谨的治学态度、宽厚的长者风范和优雅的人格魅力，以及言传身教将使我终生受益。

感谢实验室的方棣棠教授、李树青教授和徐明星副教授、邬晓钧老师的良好建议，感谢实验室同学刘建、邓菁、李净、熊振宇、孙辉、刘林泉、吴根清、张国亮等在工作中的讨论和帮助，以及梁奇、钟良伍、宋战江、周迅溢等的热情鼓励和支持！

本课题承蒙北京得意音通技术有限责任公司的合作，特此致谢。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1980 年 1 月出生于福建省莆田市。

1998 年 9 月考入北京邮电大学计算机学院计算机专业，2002 年 7 月本科毕业并获得工学学士学位。

2002 年 9 月考入清华大学计算机科学与技术系攻读计算机应用硕士至今。

发表的学术论文

[1] 陈德锋，郑方，吴文虎等．动态调整直方图剪枝 PDA 声控拨号器的应用与实现．电声技术，已录用.

[2] Zheng Fang, Chen Defeng, Wu WenWu, et al. The Dynamically-Adjustable Histogram Pruning Method for Embedded Voice Dialing. The International Association of Science and Technology for Development (IASTED), Signal and Image Processing (SIP2005), submitted.