分类号 TP391 密级 公开

重庆邮电大学硕士学位论文

论文题目 语音识别系统噪声鲁棒性算法研究

英文题目 Research on Noise Robust Speech Recognition

硕士研究生 蒲 甫 安

指导教师 李银国 教授/郑方 教授

学科专业 计算机应用技术

论文提交日期 2012 年 4 月 论文答辩日期 2012 年 5 月

论文评阅人

答辩委员会主席

2012 年 5 月

摘 要

近年来，随着语音识别技术不断地发展，语音识别系统已开始从 PC 机逐步走向嵌入式平台。然而当嵌入式语音识别系统应用到真实的操作环境中时，由于训练环境和识别环境的不匹配，导致其识别性能大大的下降。本文的重点是针对这些问题，对语音识别系统噪声鲁棒性算法展开研究。论文的主要工作有如下几个方面：

第一，构建了一个非特定人语音识别的仿真系统，系统采用一个简化的连续隐马尔科夫模型，即非线性分段与高斯混合模型（NLP+GMM）。该系统将用于噪声鲁棒算法的研究与测试。随后以该系统为基础，进行了谱减法（SS）和维纳滤波（WF）的语音增强实验。实验结果表明，在低信噪比情况下，两种语音增强算法都使系统对噪声的鲁棒性得到提升。

第二，提出了求取倒谱均值方差归一（CMVN）参数的递推算法。该递推算法能在线性时间复杂度内计算出均值和方差，使得 CMVN 参数的平均计算速度得到显著提升。

第三，在 CMVN 算法基础上，提出了基于统计阈值的 CMVN，即 STCMVN 算法。STCMVN 算法不仅能滤除特征空间的高频噪声，而且进一步减小训练环境和识别环境的不匹配。实验表明，在信噪比较低情况下，STCMVN 都要优于 MFCC、CMS 和 CMVN；CMVN 与 MFCC 相比，相对提升率最高达到 24.03%，而 STCMVN 与 CMVN 相比，相对提升率最高达到 3.03%。

第四，提出了语音增强与特征变换的两种融合算法。算法一：只将增强后的语音应用于 VAD，而特征提取使用原始带噪语音；算法二：将增强后的语音既用于 VAD 又用于特征提取。实验表明，两种融合算法的识别结果整体上都要好于文中未融合噪声鲁棒算法的识别结果。就这两种融合算法相比较而言，在较低信噪比（$SNR \leqslant 5dB$）情况下，算法一的识别率高于算法二；在较高信噪比（$SNR \geqslant 10dB$）时，算法二的识别率高于算法一。

关键词：噪声鲁棒性，语音识别，嵌入式，语音增强，特征变换

Abstract

In recent decades, with the continuous development of automatic speech recognition technologies, the speech recognition systems have gradually been applied in embedded platform from personal computer in lab environment. When the embedded speech recognition systems are applied to the real operating circumstances, however, a mismatch between training and testing conditions often causes a drastic decrease in the performance of the recognition system. In this thesis, noise robustness algorithms of speech recognition are studied. The main works of this thesis described as follow:

First, speaker-independent speech recognition system is implemented, adopting a simplified Hidden Markov Model (HMM), that is, Non-Linear Partition and Gaussian mixture model (NLP+GMM). This system will be used for research and testing of the noise robustness algorithms. Subsequently, the speech enhancement experiments are performed on this system. The experimental results show that the recognition rates of the two speech enhancement algorithms, Spectral subtraction (SS) and Wiener filtering (WF), are raised at lower SNRs.

Second, a recursive way is proposed to obtain segmental Cepstral Mean And Variance Normalization (CMVN) parameters. The average time of this approach to get the parameters is improved significantly in the liner time complexity.

Third, a Statistical Thresholding on CMVN (STCMVN) approach is proposed, which not only filters out high frequency noise in feature domain, but also further reduces the mismatch between training and testing environments. Experiments indicate that the proposed approach outperforms other robust features in lower SNR conditions, with a relative increase rate 3.30% comparing with CMVN, which has a relative increase rate 24.03% comparing with MFCC.

Lastly, the fusion of speech enhancement and feature transform is proposed and applied in speech recognition system. There are two ways to fuse the two types of the noise robust algorithms. Method 1 is that the enhanced speech is only used in VAD, and the raw noise speech or unprocessed speech is used for feature extraction. Method

2 is that the enhanced speech is both used in VAD and feature extraction. The experimental results show that the two fusion methods outperform other methods in all levels of SNRs. But the two methods each have advantages and disadvantages of various, mainly as follows: Method 1 is slightly higher than the Method 2 at higher SNR; in lower SNR, Method 2 is higher than Method 1.

Key Words: Noise Robustness, Automatic Speech Recognition, Embedded System, Speech Enhancement, Feature Transform.

目 录

第1章 绪论

从人类史前文明到如今的数字媒体时代，语音交流已成为人类社会形成和信息交流的主导模式。语音不仅是语言声音的表现形式，而且还是人类特有的自然属性之一。在人类众多的交流沟通方式中，语音毫无疑问是最自然、最有效、最直接的。没有语音的沟通，信息交流就受到阻碍；信息不能得到流畅的交流，人类社会的形成和发展也就无从谈起。

当人类进入瞬息万变的信息时代时，计算机已经成为我们生活中必不可少的一部分。随着计算机技术的发展，人们不再满足于让计算机继续做一些简单的计算，而是向它提出了更高的要求——向智能化方向发展。人们更加期待让机器明白自己在说什么，更加期待人机之间能够进行更自然的交流。在这种情况下，如何让计算机听懂人类的语言，让人机之间的交流更加自然，便成为当今研究的热门领域。语音识别就是这样一种技术，在任何情况下，机器通过识别和理解过程，把人类的语音信号转变为相应的文本或命令，其最终目标是实现人与机器进行自然的语言通信。语音识别是一门交叉学科，它所涉及的领域包括信号处理、模式识别、概率论和信息论、发声机理和听觉机理、人工智能等。

本章首先介绍本文相关研究的背景和现状，最后是本文内容结构的安排。

1.1 语音识别技术研究现状

语音是语言信息的载体。语音识别最基本的任务是将输入的语音信号转化为相应的语言符号。这不仅使得存储或传输语言符号的数码率比存储或传输原始语音信号的数码率大大的降低，而且还将连续多变的语音数字信号转变成一种有限的符号。这样得到的有限符号很容易被计算机识别处理，并理解其含义便于与人进行交流，因而语音识别技术得到广泛的研究[1-5]。

语音信号中包含了许多有意义的信息，主要包括以下几个方面：

1. 音韵信息，即同一发音的共性特征；

2. 音律信息，即有关个人特征的信息，如音强、节奏、音高等；

3. 语言信息，即说话人使用何种语言，如英语、汉语等；

4. 方言信息，对于一种语言可能有若干种不同的方言（也称口音），如四川话，普通话、粤语等；

5. 情感信息，即语音中带有说话人的情绪。

从广义上来说，语音识别也包括了说话人识别[6][7]、方言识别[8]、语言识别、

情感识别等，但在本文中主要研究的是有意义、有内容的识别，即音韵信息的识别。

1.1.1 语音识别系统的分类

根据识别对象不同，语音识别的基本任务大体可分为 3 类[9]，即孤立词识别（Isolated word recognition），关键词检出（Key words spotting）和连续语音识别（Continuous Speech Recognition）。其中，孤立词识别的识别单元为字、词或短语，如“开机”、“关机”等，由它们组成识别的词汇表，对他们中的每一个通过训练建立标准模板或模型；连续语音识别的任务则是识别任意的连续语音，如一个句子或一段话；关键字检出的输入也是连续语音流，但它并不识别全部文字，而只是检测已知的若干关键词是否在句子中出现以及在何处出现，如在一段话中检测“计算机”、“世界”这两个词。

根据针对的发音人，可以把语音识别技术分为特定人语音识别（Speaker-Dependent）和非特定人（Speaker-Independent）语音识别。特定人语音识别的标准模型或模板只是用于某一个人，实际上，该模型就是该人通过词汇表中的每个字、词或短语的语音建立起来的。当其他人也需要使用时，需要建立自己相应的标准模型。而对于非特定人的语音识别，其模型适用于指定的某一范畴的说话人（如说标准普通话），其模型是由该范畴的多个人通过训练他们的语音而得到的，识别时可以供参加训练的发音人（集内）使用，也可以是未参加训练的却在同一范畴的发音人（集外）使用。显然，非特定人语音识别系统更符合实际需要，但它要比针对特定人的识别困难得多。

另外，根据语音设备和通道，可以分为桌面语音识别、电话语音识别和嵌入式设备(手机、PDA 等)语音识别。不同的采集通道会使人的发音的声学特性发生变形，因此需要构造各自的识别系统。

1.1.2 语音识别技术基本问题

尽管语音识别的研究工作迄今已近 60 年，但仍未有突破性进展，主要原因如下[10]：

1. 语音识别系统的适应性差。一方面全世界有近百种官方语言，每种语言有多达几十种方言，同种语言的不同方言在语音上相差悬殊，这样，随着语言环境的改变，系统性能会变得很差。另一方面不同的说话人或说话的方式不同也会造成影响，如朗读式发音、随意发音和说话语速会对识别模型的结果造成影响。

2. 应用环境、采集设备和传输信道的不同。由于语音数据大部分都是在接

近理想的条件下采集的，语音识别的编码方案在研制时都要在高保真设备上录制语音，尤其要在无噪环境下录音。然而，由这些语音经训练得到的声学模型，在走向实际应用环境时，由于环境噪声的存在所带来的问题就变得越来越重要。该问题是本文研究的重点。

3. 语音信号和自然语言的多变性和复杂性。联系语音词与词之间停顿不明显，使得词与词之间的分割比较困难，同时每一个基本的声学识别基元（如音素）受前后音素发音方式的影响（也称为协同发音），使特征变得非常不稳定。对于不同人、不同生理和心理特征在不同说话环境下说同一词时，声学特征也会发生变化。自然语言的多变性难以用一些基本语法规则进行描述，因而增加了计算机编程的困难。

4. 体态语言难以识别。有人在讲话时习惯用眼神、手势、面部表情等动作协助表达自己的思想。由于这种体态语言的含义与个人习惯、文化背景、宗教信仰及生存地域等因素有关，其信息提取非常困难。

5. 对于人类由中枢神经控制的记忆机理、听觉理解机理、联想判断机理等，人们目前仍知之甚少。

1.2 嵌入式语音识别技术

1.2.1 研究意义与难点

语音识别技术发展到现在，主要有两个大的运用方向。其中一个方向是大词汇量连续语音识别系统，主要应用于计算机听写机、电话网或者 Internet 相结合的语音查询信息服务系统，这些系统都是在 PC 机平台上实现。另外一个方向是小型化、便携式、移动化和终端化的智能设备[11][12]，即嵌入式平台上的语音产品，如智能手机上的语音拨号、汽车设备的语音控制、智能玩具、家电声控设备等，这些应用系统大部分都使用专门的硬件系统实现。

随着科学技术不断的发展，移动信息时代、嵌入式时代的来临，人类越来越需要和这些智能设备进行交互，尽管交互的方式多种多样，但毫无疑问的是使用自然语言显然最为便捷。在这个背景下，语音识别系统开始从普通 PC 平台走向智能设备、嵌入式平台等。

然而由于语音识别算法的复杂性、庞大的词汇库、应用环境以及嵌入式平台的各种受限资源都制约了嵌入式语音识别技术的发展。因此如何构建出体积小，耗电省，价格低，便携性好，可支持移动作业并能适应各种复杂环境的嵌入式语音识别系统成为当前的一大研究热点。

1.2.2 研究现状

由于资源的限制，在当前的嵌入式语音识别系统多为中、小词汇量的语音识别系统，即只能识别 10 至 100 个词条[13][14]。而且该系统一般仅局限于特定人语音识别的实现，即需要让使用者对所识别的词条先进行学习和训练，这一类识别系统对词条、语种以及方言没有什么限制。由此芯片组成一个完整的语音识别系统。因此，除了语音识别功能以外，为了有一个好的人机界面和识别正确与否的验证，该系统还必须具备语音提示（语音合成）及语音回放（语音编解码记录）功能。多为实时系统，即当用户说完待识别的词条后，系统立即完成识别功能并有所回应，这就对电路的运算速度有较高的要求。除了要求有尽可能好的识别性能外，还要求体积尽可能小、可靠性高、耗电省、价钱低等特点。

1.3 语音识别的噪声鲁棒技术

在早期的语音识别研究中，大多数情况下标准数据库都是在相对安静的环境录制的，这样训练得到的系统，虽然在相同环境下可以获得很高的识别率，但是如果在实际带有噪声的环境下测试，其性能往往会变得非常差，主要原因就是带噪语音特征分布和声学模型分布之间的差异所导致[15]。

语音识别系统的噪声鲁棒性以噪声为研究对象，主要目的是减少由噪声造成的训练环境和实际应用环境的不匹配，这里的噪声包括背景噪声和信道噪声。由于语音信号和实际噪声这两者在统计上都是极其复杂的，所以噪声鲁棒性至今也没有完美的解决方案，但是在某些受限环境下，我们可以有针对性的加以解决。

1.3.1 噪声对语音识别性能的影响

大量实验表明[16-19]，在大多数现有非特定人的语音识别系统中，当训练使用的麦克风与识别使用的不相同时，识别性能都会严重下降。而对于汽车、街道、餐馆、商场、飞机、人群等环境中的语音来说，现有识别系统的鲁棒性变得更差。

在基于统计模型的语音识别系统中，训练数据必须要具有充分的代表性。但当识别系统应用于噪声环境时，纯净的训练数据与真实环境中被噪声污染的测试数据存在着不匹配，正是这种不匹配使得识别系统在噪声环境下的性能大大的下降。

由噪声造成的训练和测试的不匹配可以从信号空间、特征空间和模型空间三个层次来分析。

1.3.2 噪声鲁棒语音识别技术综述

噪声环境下的语音识别一直是一个研究热点，也称作噪声鲁棒语音识别技术。到目前为止，噪声鲁棒技术层出不穷，主要围绕信号空间、特征空间和模型空间三个方面。

1. 信号空间的噪声鲁棒技术

信号空间的噪声鲁棒技术主要关注于对原始语音信号的处理，主要包括端点检测和语音增强两方面。

端点检测(EPD, Endpoint Detection)也称语音激活检测(VAD, Voice Active Detection)。其主要目的是从麦克风采集的数字信号中区分出语音信号与非语音信号，这有利于减少非语音信号对语音识别系统的干扰，从而减少识别时间和提升识别性能。传统的端点检测方法有基于能量的和基于过零率的[20]，但这些方法在较大的噪声环境中，其性能开始恶化，不能很好的区分语音和噪声，特别是有些清音和噪声的特点相似，根本检测不出来。之后在基于传统的方法基础上，根据不同的应用需求又提出了许多新的方法[21]，包括基于基频[22]、对数能量等。这些方法将在本文的第二章进行简单介绍。

语音增强的目的是尽可能地从带噪的语音信号中提取出原始的纯净语音信号。由于不同的噪声具有不同的特性，所以不存在一种可以通用于各种背景噪声环境的语音增强算法。基于短时谱估计方法是语音增强最常用的一种方法，主要包括谱减法[23][24]、维纳滤波等。需要注意的是语音增强算法在去除噪声的同时，会残留下一些非常刺耳的音乐噪声，从而造成原始语音信号失真。为了抑制音乐噪声对语音信号的影响，一些文献提出了时域和频域的平滑方法[25-27]。

2. 特征参数空间的噪声鲁棒技术

特征参数空间噪声鲁棒技术的主要目的是在声学特征层减小训练和测试的不匹配所带来的影响，包括鲁棒性特征提取，特征归一化等。

鲁棒性特征提取主要是研究人类语音具有的特性，试图选择对噪声不敏感的特征参数。这种方法的优点是假设噪声的影响很小，并且利用了人的生理特性和听觉特性，所以适用于大部分噪声环境；缺点是没有充分地利用特定噪声的性质。这种方法包括基于人耳听觉特性的鲁棒性特征选择方法，如 MFCC [28][29]和 PLP[30]；基于人类声道特性的鲁棒性特征选择方法，如 LPCC。

特征归一化方法也称特征规整、特征后处理等，是指在提取声学特征后，通过对特征的归一化处理或者进行某种变换，将特征从一个空间变换到另一个空间，这个过程不需要太多的声学知识。特征参数归一化的主要作用有：变换后的特征参数更加符合某种概率分布、压缩了特征参数值域的动态范围、减少了训练

和测试环境的不匹配等。常用的特征归一化方法[31][32]有倒谱均值减(Cepstrum Mean Subtraction, CMS)，倒谱特征均值方差归一(Cepstrum Mean and Variance Normalization, CMVN)等。其中 CMS 能简单有效地降低了卷积噪声的影响；CMVN 继承了 CMS 的特点，不仅对卷积噪声有很好的效果，而且还能提升其对加性噪声的鲁棒性。特征参数归一化方法原理简单、计算量小，非常适用于计算资源受限的系统。

3. 模型空间的噪声鲁棒技术

模型空间的噪声鲁棒技术主要方法是通过调整已经训练好的模型参数来减小声学环境的不匹配，它包括模型补偿和模型自适应技术。

模型补偿是直接在识别模型中增加对环境噪音的处理。最具代表性的方法是平行模型合并 (Parallel Model Combination, PMC) [33]。

自适应技术的任务是让纯净语音的模型参数在不同的环境下具有一定的自适应能力，即能根据当前环境中的噪声情况自动更新模型参数，以提高系统在该噪声环境下的识别性能。主要方法有雅克比自适应技术(Jacobian Adaption)[34]和最大似然线性回归技术(Maximum likelihood Linear Regression, MMLR)[35]等。

尽管这三种方法都各具有各自的优点，但它们也有各自的不足。

信号空间级噪声鲁棒技术的主要缺点有：(1). 清辅音和宽带噪声很难区分且清辅音的相对失真比浊辅音和元音要大，一方面是因为清辅音的能量较小；另一方面是因为清辅音和宽带噪声在频谱上具有非常大的相似性，使得两者不容易区分。(2). 信号级方法在去噪后会残留下一些音乐噪音，当信噪比越大时该现象就越明显，从而造成语音再度的失真，因此许多系统仅采用增强后的语音作端点检测，特征提取选择原始带噪语音或未经处理的语音（Unprocessed Speech）。

特征参数级噪声鲁棒技术的缺点主要有：(1). 对于鲁棒性特征参数提取而言，目前的方法都是从现象入手，语音的本质特征并没有完全体现出来，如 MFCC 从人类听觉感知入手。(2). 绝大部分噪声都是非平稳噪声，因此其时变性很强，使得噪声的特性很难得到运用。(3). 对特征参数变换法而言，由于目前常用的特征参数与人的听觉机理没有密切关系，听觉上失真小并不能保证识别效果好。

模型级噪声鲁棒技术的缺点主要是所使用的自适应处理仅针对噪声模型的自适应，而不应该对其它非噪音的语音基元模型使用。而且这种方法计算量较大，对计算机的处理性能有所要求，不适用于快速改变的环境。

1.4 论文研究思路和结构安排

本文研究的对象是噪声鲁棒语音识别技术，主要目的是减少噪声或噪声处理对语音识别系统的性能影响，最终目标是能将这些技术运用到嵌入式语音识别系

统中。

首先，研究了目前常用的噪声鲁棒性算法，并通过对比各种算法在不同噪声环境下的准确率和识别效率。综合各种算法的优缺点，选择出既能适用于资源受到限制的嵌入式平台，又能满足在噪声环境下具有较好鲁棒性的算法。

其次，在提升识别效率方面，主要的考虑是对算法进行优化，利用算法自身具有的特性，并根据嵌入式系统的具体运用对某些需要运算得到的参数采用查表法代替；采用参数较少的连续统计模型替代离散的模型；利用算法中已经计算的结果来递推新的计算，从而优化算法结构；采用静态内存分配，尽管这种方法降低了内存利用率，但却减少了动态内存分配时所带来的时间开销以及内存碎片等问题。

本论文主要内容安排如下：

第 1 章为绪论，首先对语音识别技术、噪声环境下的语音识别技术和嵌入式语音识别的研究背景、相关概念以及研究现状进行综述。明确了语音识别系统噪声鲁棒性研究的背景和意义，并指出了它们在嵌入式平台上的运用前景，简要描述了现有的工作和存在的挑战，最后介绍了论文的主要工作。

第 2 章介绍了语音识别技术和噪声鲁棒技术的一些常用基本技术，语音识别技术包括语音识别的基本框架、语音信号的特点、端点检测技术、特征选择技术、声学特征建模等。其中着重介绍了高斯混合模型和非线性分段技术，因为它们主要针对嵌入式语音识别。噪声鲁棒性技术包括语音增强和特征变换。

第 3 章构建了一个基于 matlab 的语音识别系统，该系统具有系统参数配置、数据读入、前端处理、模型训练以及噪声鲁棒算法测试等功能。该系统有助于指导嵌入式语音识别系统的实现，并方便计算量较大的噪声鲁棒性算法测试。在本章中还包括前期的语音数据和噪声数据的准备。最后，给出了语音增强算法的实验结果与分析。

第 4 章中首先提出了快速 CMVN 的递推算法，并对递推算法进行分析和实验；其次提出了基于统计阈值的倒谱均值归一，并在理论和实践上对该算法进行检验。最后是特征参数各种变换算法的实验结果和分析。

第 5 章主要是语音增强技术和特征变换技术两者的融合，本章中主要介绍了它们的两种融合算法，并对这两种融合算法进行实验和分析。

第 6 章是总结与展望，总结论文的主要研究工作，指出其中的不足，并展望之后的研究工作。

第2章 语音识别技术与噪声鲁棒性技术

语音识别是一门新兴边缘学科，它主要研究如何从语音数字信号中提取最基本、最有意义的信息，它是语音数字信号处理学科的一个分支。语音识别所涉及的学科领域包括数字信号处理、物理学（声学）、模式识别、通信及信息理论、语言语音学、生理学（人类发音机理）、计算机科学（研究软硬件算法以便更有效地实现用于识别系统中的各种方法）、心理学等。在本章中将系统介绍语音识别技术和噪声鲁棒性技术的基本概念、原理、方法和应用。

2.1 语音识别技术

2.1.1 语音识别基本框架

不同的语音识别系统，虽然具体实现细节有所不同，但所采用的基本技术相似，一个典型语音识别系统的实现过程如图 2.1 所示。

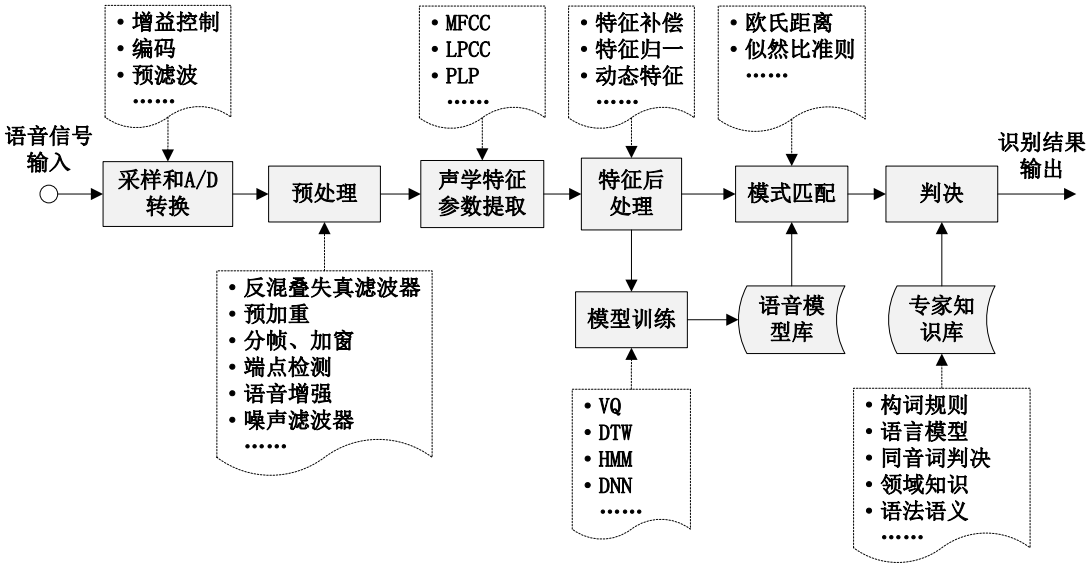

图 2.1 语音识别基本框架

首先，麦克风将接收到的待识别语音转换成电信号，并经过预增益控制、滤波采样、模数转换以及编码等过程得到语音的数字信号。这时该数字信号需要进行预处理，预处理包括反混叠失真滤波、预加重，分帧，加窗，端点检测等，必要时，还可以在此环节中增加语音增强和噪声滤波器等信号空间的抗噪技术。

经过预处理后，按照一定的特征提取方法获得语音的声学特征参数，这些特

征参数的时间序列便构成了输入语音的特征序列。在之后的特征后处理过程中可以对声学特征提取其动态特征，也可以进行特征补偿和特征参数归一化等处理。

当特征参数序列进入模型训练模块中通过不同的训练模型可以得到相应的声学模型，并存入语音模型库（也称参考模型库）中。当特征参数序列进入模式匹配模块时应根据不同的声学模型选择不同的度量准则，当声学模型为 VQ、DTW，DHMM 等时，应使用欧氏距离度量准则；当声学模型为统计模型时，应选择似然比为度量准则。经过模式匹配后得到待识别特征和模型间的距离或似然分。

最后根据模式匹配得到的距离值和似然分，并结合专家知识库中的语言模型、构词规则、领域知识、同音词判别、语法语义等进行判别，得到最终的识别结果。

2.1.2 语音的收集和预处理

2.1.2.1 预滤波、采样和量化

为了将物理波形态的语音转换成数字信号，必须经过预滤波（Pre-filtering）、采样（Sampling）和量化（Quantization），从而得到时间和幅度均离散的语音数字信号。

预滤波的主要目的有两个方面，一方面抑制输入信号中频率超过 $f_s/2$ 的分量，以防止混叠干扰，其中 f_s 是采样率；另一方面是减少 50Hz 的交流电频率干扰。这样预滤波器便是一个带通滤波器。

采样是在采样脉冲的作用下，将时间上、幅值上都连续的模拟信号转换成时间上离散（时间上有固定间隔）、但幅值上仍连续的离散模拟信号。所以采样又称为波形的离散化过程。每秒钟的采样样本数称为采样频率。采样频率越高，数值化后的声波就越接近原始的声音波形，即声音的保真度也就越高，但由于采样样本的增多，便会对传输速率和存储造成压力。根据采样定理，只有当采样频率高于声音最高频率的两倍时，才能把离散数字信号表示的声音信号唯一地还原成原来的声音[36]。因此，采样频率决定了声音频率的范围。一般而言，PC 的语音识别系统采样率为 16kHz，嵌入式平的为 8kHz。

预滤波和采样之后要对信号进行量化，即 A/D 转换。量化是将采样得到的离散点的值用二进制表示以方便计算机传输、运行和存储。常用的量化方法是将整个幅度值区间等间隔的划分，并用一个固定的离散点表示，称为量化电平。每一个语音采样之后的数据点用其所在区间对应的离散点或量化电平替代。存储时可以采用简单的二进制编码方案，即如果量化电平的个数为 256，则可以使用 8 位（bit）二进制来进行编码。这种方法被称为均匀量化，编码方法称为脉冲编码

调制（Pulse Code Modulation，PCM）。在当前的语音处理系统中，常用的编码位数为 16，即经常所说的“16 比特量化”。

实际上，预滤波、采样、量化等功能都可以用同一块芯片来完成。

2.1.2.2 语音信号的短时分析技术

语音数字信号处理处理的前提和基础是语音信号分析，只有通过分析才能找出语音信号的本质特性，才有可能利用这些参数进行高效的语音编码、语音合成以及语音识别等处理。而且语音识别效率的高低、语音合成的音质好坏都依赖于语音信号分析的准确性和精准性。因此语音信号分析在语音数字信号处理中起至关重要的作用。

目前，语音分析的全过程中最常用的技术是“短时分析技术”[37][38]。语音信号是一个非稳态过程，因为它总是随时间的变化而变化。另一方面，从人的发声基理来说，人的发声是由人声带振动产生的激励信号和声道不同形状对该激励信号的响应而形成的，声道的不同形状由口腔的肌肉运动来控制，而这种口腔肌肉运动相对于语音的频率来说是非常缓慢的。

因此，尽管从整体上看语音数字信号具有时变的特性，但在一个短时间间隔内其特性基本保持不变即相对稳定，所以可以看成一个准稳态过程，即语音具有短时平稳特性，如图 2.2 所示。

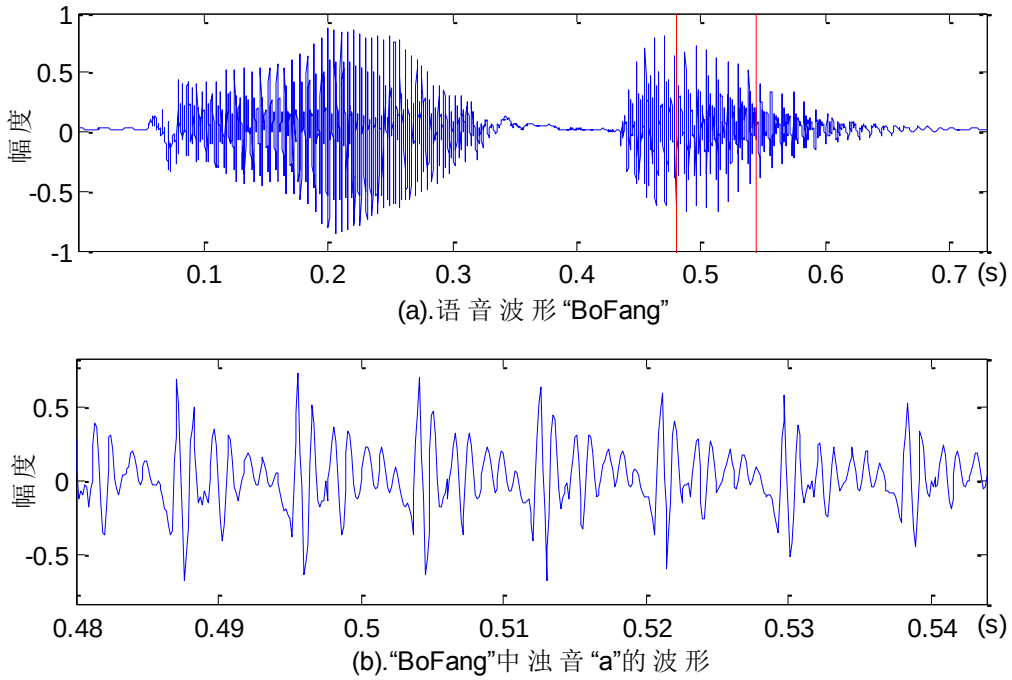

图 2.2 语音信号的短时平稳特性

语音信号的短时分析一般是将一段语音分成若干种连续的帧，每帧的长度一般取为 10~30ms，这样对语音信号的分析就是对这些帧的分析，一段语音信号的

特征参数便是由这些帧的特征参数按时间序列组成的特征序列。

事实上，在大多数应用中端点检测、语音增强等也是以短时分析技术为基础，即以帧为基础。

2.1.2.3 预加重

产生语音信号的过程是一个比较复杂的过程。总体来说是由声源去激励声道，由声道调制后，再由口鼻向外进行辐射输出。激励可以定性地分作两类：第一类是准周期脉冲串，用于激励声道以产生浊音；第二类是随机噪声，具有白噪声性质，用于激励声道以产生清音。

有研究表明，声带产生的脉冲波形与斜三角形的脉冲类似，可以看成是加权后的单位脉冲串激励经过了一个二极点斜三角波模型。另一方面，口鼻的辐射效应可以看成是一个一阶的类高通滤波器。受到这两种影响，使得输出语音信号的能量在高频部分有所下降。因此，在进行实际语音信号分析时，常常在预处理中进行预加重（Pre-emphasis）处理。其目的就是提升高频部分，使语音信号的频谱变得平坦，保持从低频到高频的整个频带中，能用同样的信噪比求频谱，以便于进行声道参数的分析或频谱分析。

预加重一般在采样和 A/D 转化之后进行。预加重数字滤波器一般是一个一阶的高通滤波器：

$$H(z) = 1 - \mu z^{-1} \tag{2.1}$$

其中 μ 的取值一般在 0.9~1.0 之间。

原始语音信号 $s[n]$ 经预加重后的信号为 $\hat{s}[n]$，括号中的 n 表示第 n 个采样点。则它们在时域上的关系为

$$\hat{s}[n] = s[n] - \mu s(n-1) \tag{2.2}$$

2.1.2.4 语音信号分帧

前面已经介绍语音信号的分析主要以短时分析（Short-term Analysis）为主，因为语音信号在短时间内是相对稳定的。短时分析的第一步就是对语音信号分帧，每一帧中含有的采样点数称为帧长（Frame Size）。相邻帧与帧之间允许重叠，前一帧与后一帧的重叠采样点数称为帧间重叠量（Frame Overlap）。当前帧的起点距相邻下一帧起点的采样点数称为帧移（Frame Step），显然帧移等于帧长与帧间重叠量之差。每一秒语音中出现的帧数量称为帧率（Frame Rate），帧率等于采样频率除以帧移。

对于采样频率 $f_s = 16\text{kHz}$ 且每一帧的时间长度为 25ms，帧间重叠量为 15ms，那么可以简单的计算出：

$$\text{Frame Size} = f_s * 25/1000 = 400$$
$$\text{Frame Overlap} = f_s * 15/1000 = 240$$
$$\text{Frame Step} = 400 - 240 = 160$$
$$\text{Frame Rate} = f_s/(\text{Frame Step}) = 100 \text{ (frames/s)}$$

在对语音分帧是必须注意以下几点：

1. 每一帧的时间长度大约是 20~30ms。若太长，将会无法提取出语音信号随时间变化的特性；若太短则不能求取出语音特性。一般而言，一帧语音中必须包含语音信号的数个基本周期。

2. 每一帧中的采样点数，即帧长通常是 2 的整数次幂。若不是，这在进行傅里叶变化时需要补零至 2 的整数次方，以便使用快速傅里叶变换。

3. 若希望相邻帧之间的变化不是太大，即帧与帧之间平滑过渡，保持其连续性，可以允许相邻帧之间有重叠，重叠的部分可以是帧长的 1/2 到 2/3 不等。但当重叠的部分越多，帧率也就越大，对应的计算量也就增加。一般每秒的帧数（帧率）大约在 33~100 帧，应根据实际的应用而定。

2.1.2.5 去直流偏移

分帧后在对语音信号进行其他处理前，一般都要进行去直流偏移（Remove DC Bias）的操作。若输入信号中包含有 50Hz 的工频干扰或 A/D 变换器的工作点有偏移时，则会对后续计算语音信号的时域参数，如短时能量、短时过零率等都会造成很大的影响。这个问题可以采用一些硬件来解决，但也可以软件编程上加以解决，就是算出每一帧的直流分量并滤除。

设语音信号在分帧处理后得到的第 t 帧语音为 $s_t[n]$（$0 \le n \le N-1$），其中 N 为每一帧的帧长，直流偏移为 DC_t，则去直流偏移后的信号为

$$\bar{s}_t[n] = s_t[n] - DC_t \tag{2.3}$$

直流偏移 DC_t 的计算方式有两种，一种是计算 $s_t[n]$ 的中位数，另一种是计算 $s_t[n]$ 的均值。

2.1.2.6 加窗

语音信号的分帧一般可采用长度有限的、可移动的窗口加权实现，也即是用一个窗函数 $w[n]$ 乘上语音信号 $s[n]$，从而得到加窗的语音信号 $s_w[n] = w[n] * s[n]$。为了获得一帧长 10ms 到 30ms 的语音，窗函数（Window Function）的函数值只在一个较短的区域内不为 0，其余部分都是 0。

截取语音最简单的窗函数是矩形窗，帧长为 N 的矩形窗定义为

$$w[n]=\begin{cases}1, & 0\le n\le (N-1)\\ 0, & \text{其他}\end{cases} \tag{2.4}$$

另外，汉明窗（Hamming Window）定义如下：

$$w[n]=\begin{cases}0.54-0.46\cos\left(\dfrac{2\pi n}{N-1}\right), & 0\le n\le (N-1)\\ 0, & \text{其他}\end{cases} \tag{2.5}$$

窗函数 $w[n]$ 的形状和长度对短时分析参数的特性有很大的影响。因此，选择合适的窗函数能更好的反映语音信号的特性。就窗函数的形状而言，比较好的选择标准是：能减小语音帧的截断效应，即使语音帧的两端能平滑过渡到 0 而不引起剧烈的变化。如图 2.3 所示，通过分析窗函数的时域波形图，可知汉明窗更能满足这一要求。

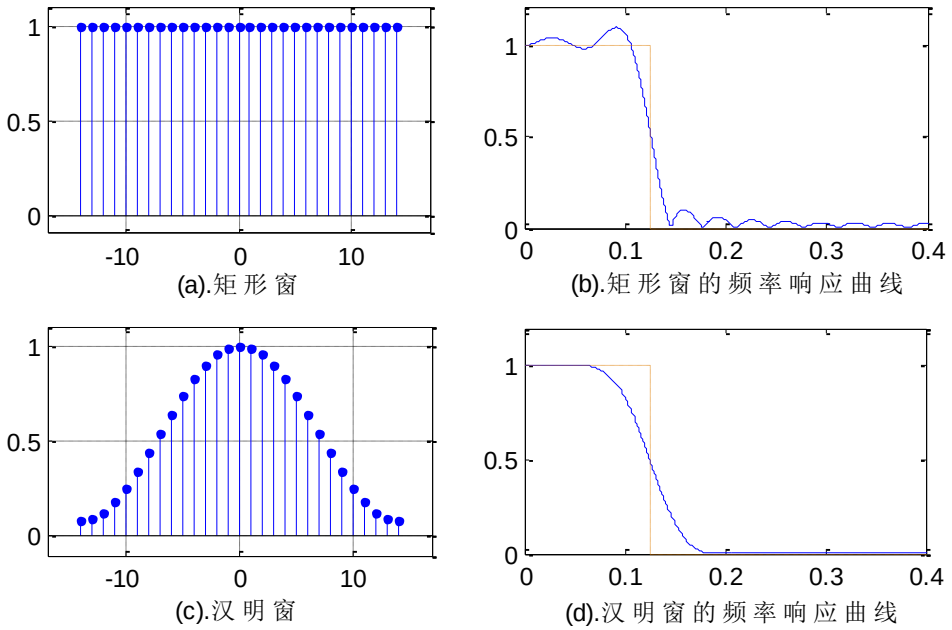

图 2.3 矩形窗和汉明窗时域波形图

在实际应用中，通常都使用汉明窗，其另一方面原因是有利于随后的傅里叶变换。在离散傅里叶变换中，会将一帧信号进行周期性的拓展形成一个周期信号，即拓展后的信号其基本周期就是加窗后的帧，则很有可能产生帧与帧之间的不连续，这样势必会造成分析误差。当然，在分帧过程中，若能使一帧中恰好包含基本周期的整数倍时，这时在进行拓展时可以保持左右连续，也就不需要乘上汉明窗了。但在实际运用中，由于基本周期的计算需要额外的时间，而且也容易计算错误，因此乘上汉明窗来达到类似的效果。

如图 2.4 所示，原始信号是对频率为 300Hz 的正弦函数加上一些白噪声后，进行 8kHz 的采样所得到的，（a）和（b）分别是对其取一帧后加矩形窗和汉明窗

后的波形图，（c）和（e）、（d）和（f）分别是矩形窗、汉明窗对应的能量谱曲线（线性尺度和对数尺度）。从图中可以看出，矩形窗能量谱曲线中的峰值两旁瓣较宽且峰值不明显；而汉明窗能量谱曲线中的峰值更尖锐且更具有区分性。

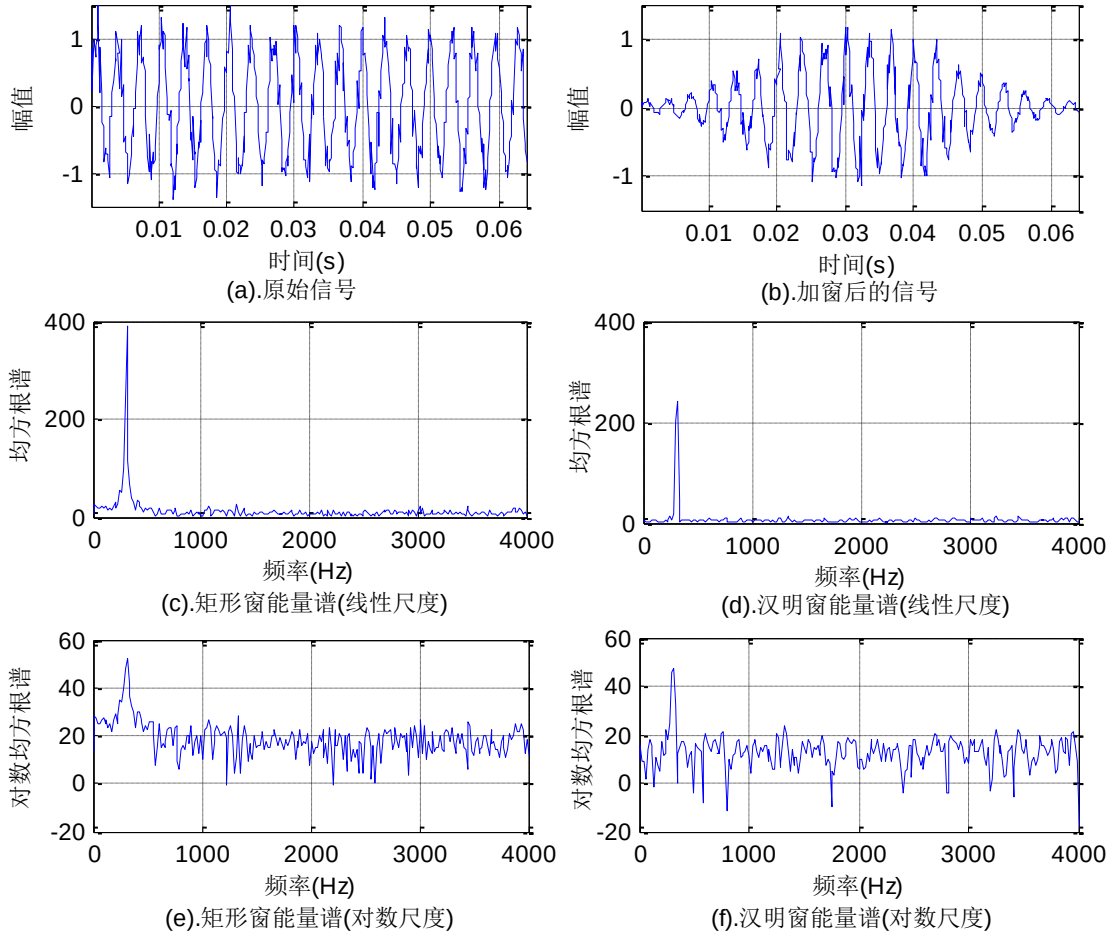

图 2.4 矩形窗和汉明窗对比

如果使用实际的语音信号测试时，汉明窗的效果会更加明显。若以图 2.2 中元音“a”的一帧语音信号为例进行同样的加窗分析，如图 2.5 所示，可以看出乘上汉明窗之后，频谱的谐波结构变得非常明显。

2.1.3 端点检测

端点检测的主要目的是检测出语音信号中语音的开始位置和结束位置，也称为语音检测（Speech Detection），或 VAD（Voice Activity Detection）。端点检测在语音信号处理和语音识别中起着重要的作用。

一般而言，常见的端点检测方法可以分为两类：时域（Time Domain）方法和频域（Frequency Domain）方法。对于时域方法而言，只是对声音波形做一些简单的运算，因此其计算量小，可以移植到计算能力差的微电脑平台上去；而对

于频域方法而言，要在频域进行分析必须先经过傅里叶变换，因此需要更多计算量，不适合计算能力较差的平台。

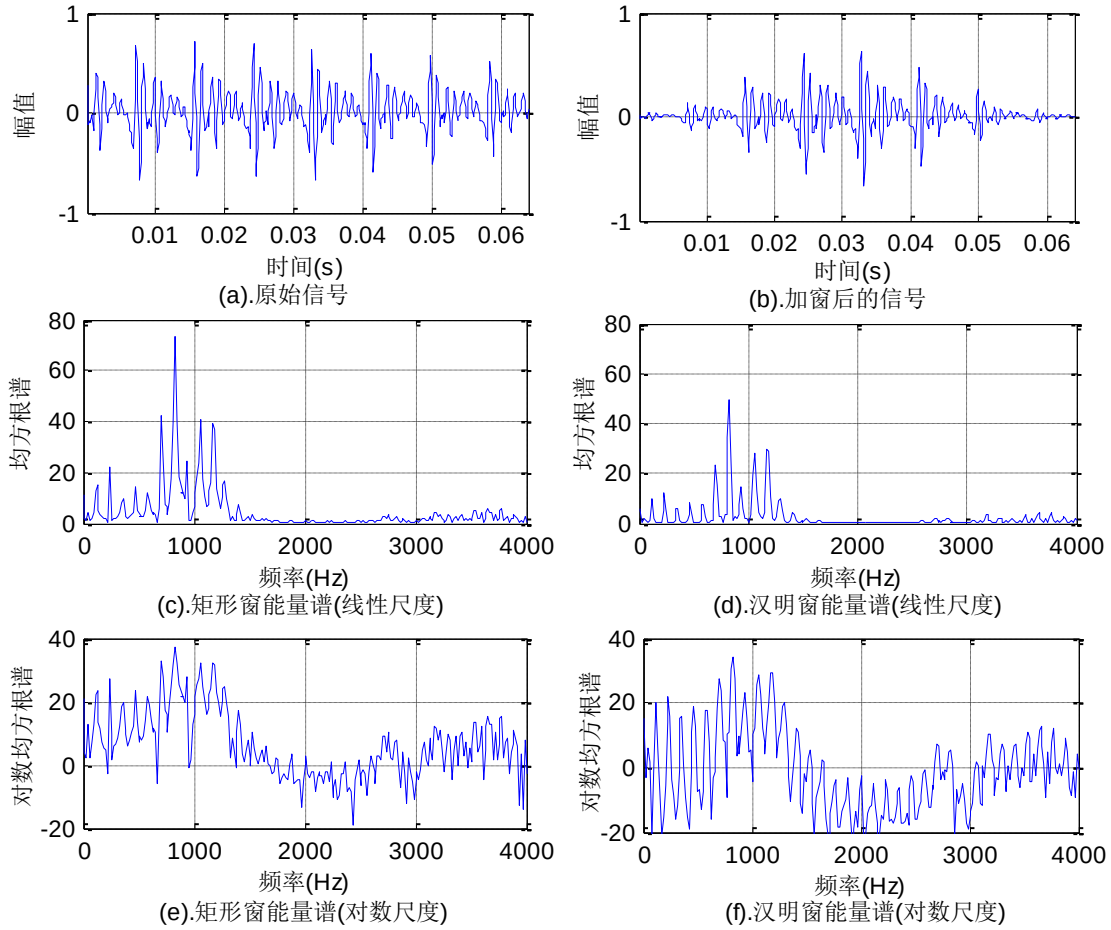

图 2.5 实际语音帧的矩形窗和汉明窗分析

错误的端点检测，在语音识别上会造成两种效应：

1. 错误拒绝（False Rejection）：将语音误认为是静音段或噪音段，因而造成识别率下降；

2. 错误接受（False Acceptance）：将静音段或噪音段误认为语音段，这时也会造成识别率的下降。但是在设计识别器时，若考虑语音段前后加上静音或噪声的声学模型（也就是认为静音和噪声是一种特殊类型的语音），此时识别率的下降会比前者来的缓和些。

由于本文跟计算能力受限的嵌入式平台相关，所以接下来将讨论时域端点检测方法中一些常用的参数和技术。

2.1.3.1 端点检测的时域参数

语音信号的时域参数，是直接利用语音的时域波形而求取的。时域参数能非常直观的表示语音信号的特性，并且具有明确的物理意义，实现起来也比较简单、

运算量少。常用的语音时域参数有短时幅度、短时能量、短时过零率等。需要注意的是计算这些参数的计算都是基于前面所说的“短时分析技术”，即以帧为基础。

1. 短时幅度、短时能量和短时对数能量

设语音信号 $s[n]$ 在加窗、分帧、去直流偏移处理后得到的第 t 帧语音信号为 $s_t[n]$，n 满足 $0 \le n \le N-1$，其中 N 为每一帧的帧长，则第 t 帧的短时幅度 M_t 计算公式如下：

$$M_t = \sum_{n=0}^{N-1} |s_t[n]| \tag{2.6}$$

同时，第 t 帧的短时能量 E_t 计算公式是

$$E_t = \sum_{n=0}^{N-1} s_t^2[n] \tag{2.7}$$

E_t 和 M_t 都具有刻画语音信号幅度值变化的特性，但对于短时能量 E_t 而言，因为计算用的是平方运算，所以对高电平十分敏感。而对于 M_t 来说，在计算小采样值和大采样值不会因取平方而造成较大的差异，并且 M_t 的计算量相对于 E_t 要少，因此若在嵌入式运用中会带来一些好处。

语音的对数能量也称短时强度（Intensity），第 t 帧的短时对数能量的 I_t 定义为

$$I_t = 10\log E_t = 10\log\left(\sum_{n=0}^{N-1} s_t^2[n]\right) \tag{2.8}$$

I_t 以分贝（Decibels）为单位，比较符合人耳对声音大小的感知[39]，是一个相对强度的值，需要更多的浮点运算。

语音信号的短时幅度、短时能量以及短时强度在很多文献中都被称作音量（Volume）。一般而言，有浊音的音量大于清音的音量，清音的音量又大于环境噪声的音量。音量是一个相对值，因为它受麦克风的增益（Gain）的影响很大。音量的主要用途是端点检测，以检测出采集信号中有意义的语音。在计算音量之前一般会进行去直流偏移操作，以避免直流偏移所导致的误差。对于短时幅度去直流偏移时一般减去其中位数，而对于短时能量和短时强度一般减去其平均值。

2. 短时过零率

过零率（Zero Crossing Rate，ZCR）是语音波形穿过 x 轴的次数，也是一种可以通过简单计算就能得出的时域参数，第 t 帧语音信号的过零率为

$$ZCR_t = \frac{1}{2}\sum_{n=1}^{N-1} |1 - \mathrm{sgn}(s_t[n-1] \bullet s_t[n])| \tag{2.9}$$

其中 sgn 是符号函数，即

$$\mathrm{sgn}(x)=\begin{cases}1, & x\geq 0\\ -1, & x<0\end{cases} \tag{2.10}$$

一般而言，过零率的特性有：浊音具有明显的周期性，因此环境噪声和清音的过零率大于浊音；环境噪声和清音非常相似，过零率很难将他们区分开来；又因为清音的音量一般大于环境噪声，因此可以将过零率和音量相结合进行端点检测。

在使用过零率需要注意的是：当环境噪声信号的值都在 0 点或者 0 点跳动，此时过零率的值会比较高，这会对端点检测的准确性造成影响；另外，如果采样率提高得到的结果也不相同。

2.1.3.2 时域的端点检测

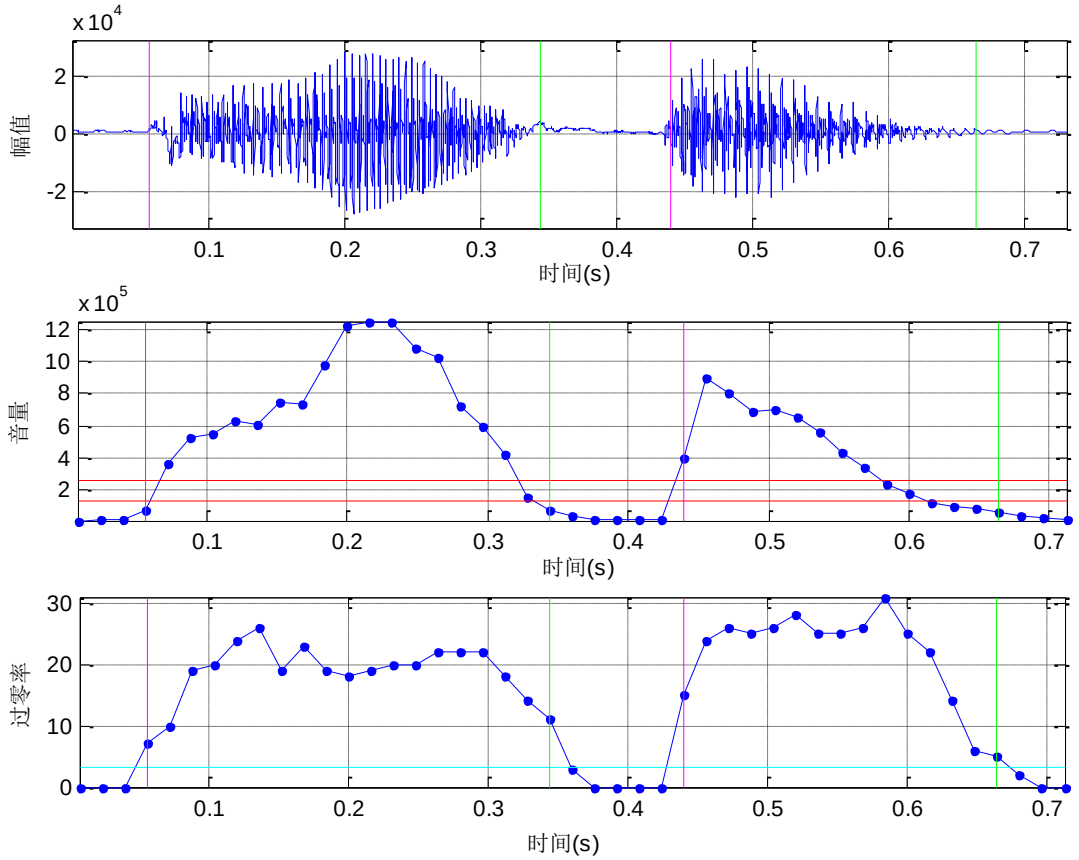

图 2.6 双门限端点检测

时域端点检测最最常用的方法是双门限法，利用过零率检测清音，音量检测浊音，两者相结合。首先为音量和过零率分别设定两个阈值，即门限值，其中可以是短时幅度、短时能量，短时对数能量中的任何一种。另外再设定一个比较高的门限。当门限被超过未必是语音的开始，很有可能由一段很短的环境噪声引起。

当高门限被超过时并且之后的几个时间段内的音量都超过低门限，便是信号的开始。

此时整个端点检测便可分为四段，也即四个状态：静音段、过渡段、语音段、结束，整个过程便是一个状态转移过程。实验时用一个变量记录当前状态。当状态是静音段时，如果能量或过零率超过低门限，就开始标记起始点，同时进入过渡段状态。若过渡段中两个参数值都回落到低门限以下，就恢复到静音状态；如果过渡段中两个参数中的任一超过高门限，则进入语音段状态。处于语音段状态时，如果两参数降低到门限以下，而且总的计时长度小于最短时间门限，则认为是一段噪音，继续扫描以后的语音数据，否则进入结束状态，并标记结束端点。如图 2.6 所示。

2.1.4 声学特征选取

提取可靠的声学特征是语音识别中最重要的问题之一。虽然有很多声学特征可供我们使用，但维数灾难（Curse-of-Dimensionality）[43]一直是一个大的问题，因为训练数据的总量经常受限。因此，联合使用额外的特征并不总能使错误率降低，但这也并不一定意味着额外的特征没什么用处，相反，若有足够的数据便能可靠地对这些特征进行建模。

语音信号最直接的特征便是语音波形本身，也即时域特征。但通常而言，时域特征的描述能力没有频域特征的描述能力强。这因为许多频域特征可以用一个低维特征向量来描述，例如共振峰（Formants）可以用来区分元音，基音（Pitch）可以用来区分说话人的性别。

在本节中将着重介绍一种常用的频域特征：梅尔倒谱特征系数（Mel-scale Frequency Cepstral Coefficients，简称 MFCC）。

2.1.4.1 标准 MFCC 特征提取

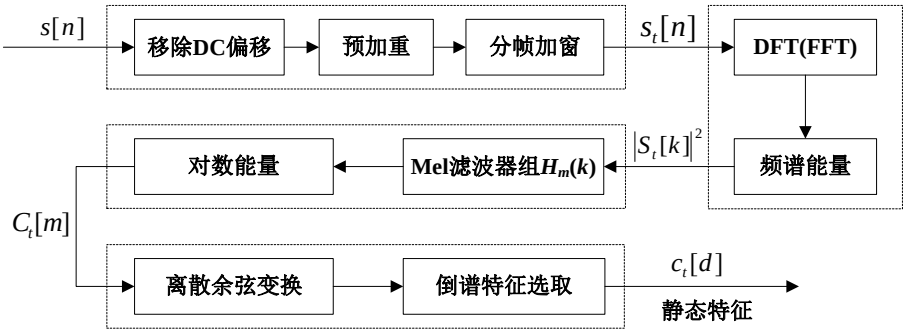

图 2.7 静态 MFCC 特征参数提取流程

MFCC 特征从人耳的听觉特性出发，考虑到人耳对不同频率的感知程度，即人耳在嘈杂的环境中，以及各种发音变异情况下仍能分辨出各种不同的声音。因此，MFCC 特征参数是一种鲁棒性的特征参数。

标准 MFCC 特征[40]也称静态 MFCC 特征，其提取流程如图 2.7 所示，它主要有 4 个步骤：

1. 首先，对输入语音信号 $s[n]$ 经移除 DC 偏移、预加重、分帧和加窗处理，得到第 t 帧语音 $s_t[n]$，其中 $0 \le n \le N-1$， N 为帧长。

2. 其次对第 t 帧语音进行 N 点离散傅里叶变换（Discrete Fourier Transform，DFT），即

$$S_t[k] = \sum_{n=0}^{N-1} s_t[n] \exp\left(-\frac{j2\pi nk}{N}\right), \quad 0 \le k \le N-1 \tag{2.11}$$

在实际应用中离散傅里叶变换通常采用 FFT（Fast Fourier Transform）替代，然后得到其频谱 $\left|S_t[k]\right|^2$。

这里使用频谱的一个主要原因是相位信息在语音识别中所起的作用并不大，人耳的感知程度与信号的能量成正比。有实验表明，当原始语音中的相位被随机相位替代时，人耳仍然能感知出语音中所包含的信息。

3. 构造 M 个三角带通滤波器组（Triangular Band-pass Filters），其中第 m 个三角滤波器 $H_m[k]$（$0 \le m \le M-1$）定义为

$$H_m[k] = \begin{cases} \dfrac{2\left(k - f[m-1]\right)}{\left(f[m+1] - f[m-1]\right)\left(f[m] - f[m-1]\right)}, & f[m-1] \le k \le f[m] \\ \dfrac{2\left(f[m+1] - k\right)}{\left(f[m+1] - f[m-1]\right)\left(f[m+1] - f[m]\right)}, & f[m] \le k \le f[m+1] \\ 0, & else \end{cases} \tag{2.12}$$

其中 $f(m)$ 为滤波器的中心频率，定义为

$$f[m] = \frac{N}{f_s} \bullet B^{-1}\left(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\right) \tag{2.13}$$

式中 f_l 和 f_h 分别是滤波器在一般频率域（Hz）的最低和最高频率， f_s 信号采样率， $B(\bullet)$ 是一般频率域到 Mel 频率域的尺度函数， $B^{-1}(\bullet)$ 是其逆函数

$$B^{-1}(f_{Mel}) = 700\left(\exp\left(\frac{f_{Hz}}{1125}\right) - 1\right) \tag{2.14}$$

使用三角带通滤波器的主要目的是对频谱进行平滑，并消除谐波的影响，突显原先的共振峰，因此一段语音的音调或基音，是不会体现在 MFCC 参数内。也即是说，以 MFCC 为特征的语音识别系统，并不会因输入语音的音调不同而

有所影响。另一方面，三角滤波器在某种程度上降低了信息量。

求得三角滤波器组后，将频谱能量通过滤波器组，得到每一个滤波器的输出，并计算其对数能量，即

$$C_t[m]=\ln\left(\sum_{k=0}^{N-1}\left|S_t[k]\right|^2 H_m[k]\right),\quad 0\le m\le M-1 \tag{2.15}$$

这里使用对数能量不仅压缩了最终取值的动态范围，以适应人类听觉特性，而且还使得提取的特征对动态变化信号的敏感程度降低。同时也使得时域的卷积经频域变换后，再到倒谱域，便变成加性，如图 2.8 所示。

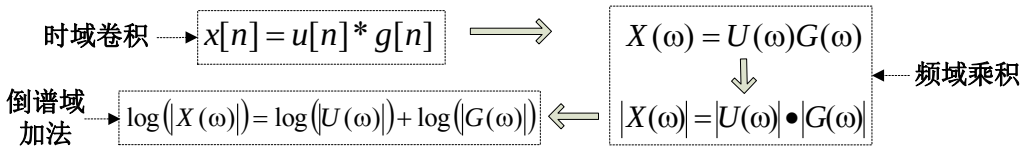

图 2.8 时域、频域、倒谱域的转换

4、对上一步各个滤波器输出的对数能量进行离散余弦变换（Discrete Cosine Transform，DCT），即

$$c_t[d]=\sum_{m=0}^{M-1}C_t[m]\cos\left(d\left(m-\frac{1}{2}\right)\frac{\pi}{M}\right),\ d=0,1,\ldots,D-1<M \tag{2.16}$$

式中，D 表示 Mel 倒谱系数的阶数，一般来说它小于三角滤波器的个数。

DCT 变换能使各维特征的相关性降低，这样便能在后续的高斯混合建模过程中使用对角协方差矩阵。

在求得最终的倒谱特征后，可以对其中的特征进行筛选，选出对噪声鲁棒的特征[41][42]，如用对数能量替代$c_t[0]$。

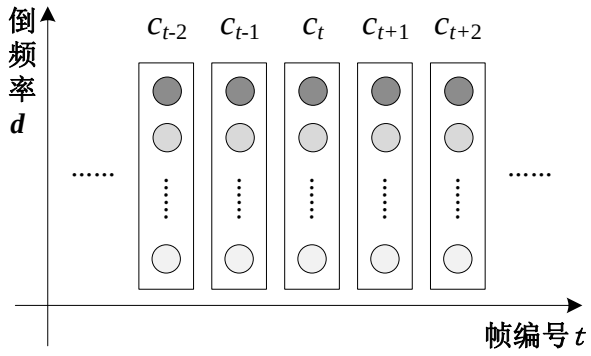

图 2.9 静态 MFCC 特征参数流

如图 2.9 所示，当输入一段语音信号时，经过以上步骤后便得到 MFCC 静态特征参数流，设第 t 帧语音求取 MFCC 静态特征后的特征向量为$\mathbf{c}_t$，则

$$\mathbf{c}_t = \left(c_t[0], c_t[1], \cdots, c_t[D-1]\right)^T \tag{2.17}$$

2.1.4.2 对数能量

在语音识别中经常使用语音信号每帧的能量作为特征参数的一个分量，这里的能量是指前面在时域参数中介绍的三种音量，最常用的是对数能量，因为对数能量更符合人耳听觉特性，能压缩能量的动态取值范围，对信号的动态变化不敏感等。当在 MFCC 特征中加入对数能量 I_t 后，这时特征向量可表示成

$$\mathbf{c}'_t = \left(c_t[0], c_t[1], \cdots, c_t[D-1], I_t\right)^T \tag{2.18}$$

2.1.4.3 动态特征参数

动态特征参数反映了特征向量随时间变化的特性，使用动态特征可以进一步提升语音识别系统的性能。研究表明，MFCC 的动态特征在一些环境下比静态特征更具有鲁棒性[29]。同时，动态特征弥补了隐马尔科夫模型（Hidden Markov Model，HMM）的不足，因为 HMM 假设了当前帧与帧之间相互独立，而动态特征正好填补了这一空缺。

若采用线性回归分析（Linear Regression Analysis，LRA）求取动态特征，则一阶动态参数计算如下：

$$\Delta\mathbf{c}_t = \frac{\sum_{i=-p}^{p} i \bullet \mathbf{c}_{t+i}}{\sum_{i=-p}^{p} i^2}, \quad (p > 0) \tag{2.19}$$

二阶动态参数是在一阶动态参数上求取，即

$$\Delta\Delta\mathbf{c}_t = \frac{\sum_{i=-p}^{p} i \bullet \Delta\mathbf{c}_{t+i}}{\sum_{i=-p}^{p} i^2}, \quad (p > 0) \tag{2.20}$$

在求取动态特征参数之后，一帧语音的特征参数便表示为

$$\mathbf{x}_t = \begin{pmatrix} \mathbf{c}_t \\ \Delta\mathbf{c}_t \\ \Delta\Delta\mathbf{c}_t \end{pmatrix} \tag{2.21}$$

在求取动态特征时需要注意的是 p 的选择， p 若太小，将越接近于中心帧，因此不能反映出语音信号的动态特征； p 若太大，则会牵涉到 HMM 模型中许多不同的状态。实验表明当 $p = 2$ 时能取得好的效果。

2.1.5 声学特征的高斯混合建模

在训练步骤中，需要对这些语音的特征参数进行建模。对离散的特征建模常用的是高斯混合模型（Gaussian Mixture Model，GMM）[43]。

2.1.5.1 GMM 的基本概念

高斯混合模型是用多个高斯概率密度函数（Probability Density Function，PDF）对同一类特征进行建模，它可以对事物进行精确量化。一个 M 阶高斯混合模型的概率密度函数由 M 个高斯分量（Gaussian component）加权求和组成，即

$$p(\mathbf{x}\,|\,\theta)=\sum_{i=1}^{M} w_i g\left(\mathbf{x},\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right) \tag{2.22}$$

其中 $\mathbf{x}$ 是 D 维随机向量，w_i（$i=1,\ldots,M$）是第 i 个高斯分量的权值，且 $\sum_{i=1}^{M} w_i=1$，第 i 个高斯分量的概率密度函数是一个 D 维高斯函数，则

$$g(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i)=(2\pi)^{-\frac{D}{2}}\left|\boldsymbol{\Sigma}_i\right|^{-\frac{1}{2}}\exp\left\{-\frac{(\mathbf{x}-\boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i)}{2}\right\} \tag{2.23}$$

式中 $\boldsymbol{\mu}_i$ 是均值，表示此概率密度函数的中心点，$\boldsymbol{\Sigma}_i$ 是其方差，表示此概率密度函数的协方差矩阵（Covariance Matrix），它们共同决定了此概率密度函数的特性，如函数形状的中心、宽窄及走向。

在式(2.22)中，参数 θ 表示一个高斯混和模型的参数集，也即是一个高斯混合模型，可表示为

$$\theta=\left\{\left(w_i,\boldsymbol{\mu}_i,\boldsymbol{\Sigma}_i\right)\,|\,i=1,\ldots,M\right\} \tag{2.24}$$

而 $p(\mathbf{x}\,|\,\theta)$ 则表示特征向量 $\mathbf{x}$ 在模型 θ 上的概率值。

在语音识别中，一个识别单元（音素、字或词等）可用一个高斯混合模型表示。S 个识别单元用需要用 S 个高斯混合模型表示，即 $\theta_1,\theta_2,\ldots,\theta_S$。当给定某个待识别单元的特征序列 $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_t,\ldots,\mathbf{x}_T\}$，其中 $\mathbf{x}_t$ 表示第 t 帧语音的特征向量，若假设每帧相互独立，则可以通过求最大后验概率来识别出待识别的单元：

$$\hat{s}=\arg\max_{j}\prod_{t=1}^{T} p\left(\mathbf{x}_t\,|\,\theta_j\right) \tag{2.25}$$

概率密度函数的取值范围都在 0 到 1 之间，因此上式中 T 帧连乘会产生下溢。在实际应用中常用对数后求和来替代，即

$$\hat{s} = \arg\max_{j} \sum_{t=1}^{T} \ln p\left(\mathbf{x}_t \mid \theta_j\right) \tag{2.26}$$

2.1.5.2 GMM 的参数估计

高斯混合模型参数估计是在给定一组数据情况下，根据某种准则确定出最佳模型参数的过程，也称高斯混合模型的训练。最大似然估计（Maximum Likelihood Estimation，MLE）是高斯混合模型最常用的参数估计方法。对于给定的特征向量训练集 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T\}$，其中 T 表示训练集的大小，则根据 MLE 准则需要求出下式的最大值

$$\begin{aligned} J(\theta) &= \ln\left(\prod_{t=1}^{T} p(\mathbf{x}_t \mid \theta)\right) = \sum_{t=1}^{T} \ln p(\mathbf{x}_t \mid \theta) \\ &= \sum_{t=1}^{T} \ln\left(\sum_{i=1}^{M} w_i g(\mathbf{x}_t; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\right) \end{aligned} \tag{2.27}$$

为简化问题，通常假设各个高斯概率密度函数的协方差矩阵都是对角元素相同的对角阵，即

$$\boldsymbol{\Sigma} = \sigma^2 \mathbf{I} = \sigma^2 \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix} \tag{2.28}$$

这时 D 维高斯概率密度函数可以表示成为

$$g(\mathbf{x}; \boldsymbol{\mu}, \sigma^2) = (2\pi)^{-\frac{D}{2}} \sigma^{-D} \exp\left[-\frac{(\mathbf{x}-\boldsymbol{\mu})^T (\mathbf{x}-\boldsymbol{\mu})}{2\sigma^2}\right] \tag{2.29}$$

定义中间变量

$$\beta_j(\mathbf{x}) = \frac{w_j g(\mathbf{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{i=1}^{M} w_i g(\mathbf{x}; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)} \tag{2.30}$$

$\beta_j(\mathbf{x})$ 可以看成随机向量 $\mathbf{x}$ 由第 j 个高斯密度函数产生的概率。为了求得 $J(\theta)$ 的最大值，可以采用求偏导数方法：

$$\frac{\partial J(\theta)}{\partial \boldsymbol{\mu}_j} = \sum_{t=1}^{T} \beta_j(\mathbf{x}_t) \left(\frac{\mathbf{x}_t - \boldsymbol{\mu}_j}{\sigma_j^2}\right) \tag{2.31}$$

$$\frac{\partial J(\theta)}{\partial \sigma_j} = \sum_{t=1}^{T} \beta_j(\mathbf{x}_t) \left[\frac{(\mathbf{x}_t - \boldsymbol{\mu}_j)^T (\mathbf{x}_t - \boldsymbol{\mu}_j)}{\sigma_j^3} - \frac{D}{\sigma_j}\right] \tag{2.32}$$

令式(2.31)和式(2.32)都为 0，得

$$\boldsymbol{\mu}_j = \frac{\sum_{t=1}^{T}\beta_j(\mathbf{x}_t)\mathbf{x}_t}{\sum_{t=1}^{T}\beta_j(\mathbf{x}_t)}，\quad \sigma_j = \frac{1}{D}\frac{\sum_{t=1}^{T}\beta_j(\mathbf{x}_t)(\mathbf{x}_t-\boldsymbol{\mu}_j)^T(\mathbf{x}_t-\boldsymbol{\mu}_j)}{\sum_{t=1}^{T}\beta_j(\mathbf{x}_t)} \tag{2.33}$$

此外，为求得高斯分量的权值 w_j，并且需要满足约束条件 $\sum_{i=1}^{M} w_i = 1$，因此使用朗格朗日乘数法，重新构造目标函数

$$\hat{J}(\theta) = J(\theta) + \lambda(1 - \sum_{i=1}^{M} w_i) \tag{2.34}$$

并对 w_j 求导：

$$\frac{\partial \hat{J}(\theta)}{\partial w_j} = \frac{1}{w_j}\sum_{t=1}^{T}\beta_j(\mathbf{x}_t) - \lambda = 0，\quad j = 1,2,\ldots,M \tag{2.35}$$

求解得

$$w_j\lambda = \sum_{t=1}^{T}\beta_j(\mathbf{x}_t) \tag{2.36}$$

式(2.36)等号两边对 j 求和

$$\sum_{j=1}^{M} w_j\lambda = \sum_{j=1}^{M}\sum_{t=1}^{T}\beta_j(\mathbf{x}_t)$$

于是可以得出

$$\lambda = \sum_{t=1}^{T}\left(\sum_{j=1}^{M}\beta_j(\mathbf{x}_t)\right) = \sum_{t=1}^{T}1 = T \tag{2.37}$$

将式(2.37)代入式(2.36)得

$$w_j = \frac{1}{T}\sum_{t=1}^{T}\beta_j(\mathbf{x}_t) \tag{2.38}$$

因此，由式(2.33)和式(2.38)便可以求得参数 θ。一般很难用直接的方法对这两式求解，通常以这两式为基础进行迭代求解，即 EM[44][45]算法。因此求得 GMM 参数的流程如下：

1. 设定初始的参数 θ。一般令 $w_j = 1/M$，并用 K-means[46][47]聚类算法将训练集分成 M 类，并用第 j 类的聚类中心和方差分别作为 $\boldsymbol{\mu}_j$ 和 σ_j 的初始值。

2. 根据参数 θ，使用式(2.30)来计算 $\beta_j(\mathbf{x}_t)$，$t = 1,2,\ldots,T$。

3. 根据式(2.33)和式(2.38)，重新计算新的参数 $\hat{\theta}$。

4. 若 $\left\|\theta - \hat{\theta}\right\|$ 小于某一非常小的容忍值，则令 $\theta = \hat{\theta}$，且算法终止。否则令

$\theta = \hat{\theta}$，并跳转到步骤(2)继续迭代。

该迭代算法一定会让 $J(\theta)$ 逐步递增，并收敛至一个局部最大值，但目前却无法证明此局部最大值是否就是全局最大值。

2.1.6 用于嵌入式平台的非线性分段与高斯混合建模

在现在语音识别系统中，隐马尔科夫模型（HMM）占主要地位。当用 HMM 作为识别模型时，特征矢量的输出概率计算和匹配搜索将占用大量的时间和空间，这使得 HMM 在资源受限的嵌入式平台上较难实现。通常一个 HMM 由状态转移概率矩阵 A，观测值概率函数 B，系统初始状态概率矢量 π。为了减少计算量和存储量，本小节将介绍一种 HMM 的简化模型[48-51]，该模型采用自左向右的状态转移结构，并且在状态转移时，状态不允许进行跨越转移，因此没有状态转移矩阵 A 和初始矩阵 π；每一个状态的特征空间用一个 GMM 描述，如图 2.10 所示。

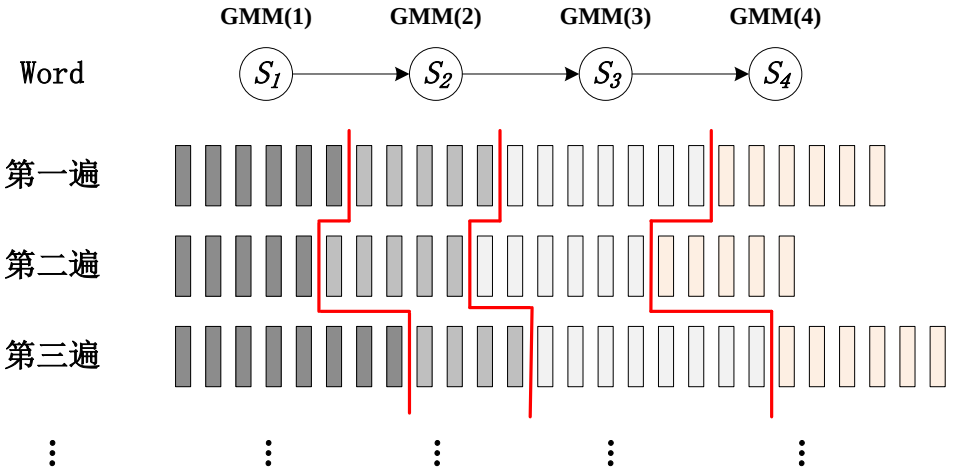

图 2.10 词的分段高斯混合建模

当没有状态转移矩阵 A 时，需要解决的问题是：如何进行特征序列的状态划分？最直接的方式是均匀划分，即每个状态中包含相等的特征个数。由于人的发音具有随意性，显然这种划分并不能准确地对每个状态的特性进行刻画。Juang 提出了一种非线性分段（Non-Linear Partition，NLP 或 Non-Linear Segmentation，NLS）[52]算法能更好地解决这一问题。

在 NLP 算法中，首先需要计算特征向量序列 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T\}$ 中相邻两个特征向量之间的距离（变化量）

$$y_t = \mathbb{D}(\mathbf{x}_{t+1}, \mathbf{x}_t)，\quad 1 \le t \le T-1 \tag{2.39}$$

其中 $\mathbb{D}(\bullet)$ 是距离准则函数[53]，可以是欧氏距离，马氏距离等。进一步可以求得

段内平均总变化量：

$$\Delta y = \frac{1}{N}\sum_{t=1}^{T-1} y_t \tag{2.40}$$

式中的 N 是状态数，即表示将特征序列分成 N 段。在图 2.10 中，每一个特征序被分成 4 段，即 N=4。

以 K_n 表示前 n 段中向量的总个数，令 $K_0 = 0$（这里使用第 0 个状态只是便于计算，而第 0 个状态实际不存在）和 $K_N = T$，而对于 $1 \le n \le N-1$，当 k 满足

$$\sum_{i=1}^{k-1} y_t < n \bullet \Delta y \le \sum_{i=1}^{k} y_t \tag{2.41}$$

则令 $K_n = k$。显然 K_n 就是第 n 个状态和第 $n+1$ 个状态的分界点，或第 n 段和第 $n+1$ 段的分界点，第 n（$1 \le n \le N$）段的起始点是 $K_{n-1}+1$，终止点是 K_n。

由 NLP 算法可以看出，每个状态段内特征变化量的总和基本一致。NLP 算法比较简单地给出了在“等特征变化量”的意义下“最好的”状态序列，而且无论状态和识别基元的驻留时间如何变化，它总能比较一致地把变化较小的那些序列划分到一起。从某种程度上说，它对变化的语速具有较好的鲁棒性。

若每一个识别单元被分成 N 段，则分别需要对这 N 段建模，若对每段用高斯混合模型进行建模，如图 2.10 所示，则一个识别单元的模型可表示为

$$U = \{\mathrm{GMM}(1), \mathrm{GMM}(2), \ldots, \mathrm{GMM}(N)\} \text{或} U = \{\theta_1, \theta_2, \ldots, \theta_N\} \tag{2.42}$$

假设模型中各状态相互独立，对于一个待识别特征序列 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \ldots, \mathbf{x}_T\}$ 而言，若需计算其在模型上的后验概率，首先用 NLP 算法计算出特征序列各个状态的分界点 K_n（$n = 0,1,\ldots,N$），然后用下列公式计算后验概率：

$$p(\mathbf{X}|U) = \prod_{n=1}^{N} p(\mathbf{X}[K_{n-1}+1, \cdots, K_n]|\theta_n) = \prod_{n=1}^{N} \prod_{t=K_{n-1}+1}^{K_n} p(\mathbf{x}_t|\theta_n) \tag{2.43}$$

式中的 $\mathbf{X}[K_{n-1}+1, \cdots, K_n]$ 表示特征矢量 $\mathbf{X}$ 的第 n 段。在实际应用中一般是对式(2.43)取对数进行计算，即

$$\ln p(\mathbf{X}|U) = \ln \prod_{n=1}^{N} \prod_{t=K_{n-1}+1}^{K_n} p(\mathbf{x}_t|\theta_n) = \sum_{n=1}^{N} \sum_{t=K_{n-1}+1}^{K_n} \ln p(\mathbf{x}_t|\theta_n) \tag{2.44}$$

若有 H 个识别单元，则需要 H 个模型表示，即 $U_1, U_2, \ldots, U_h, \ldots U_H$。对于待识别的特征序列 $\mathbf{X}$，则可以通过最大后验概率找出识别的单元，即

$$\hat{h} = \arg\max_h \ln p(\mathbf{X}|U_h) \tag{2.45}$$

2.2 噪声鲁棒语音识别技术

语音识别系统在实际应用中需要解决的一个关键问题是模型训练和应用环境的不匹配。在实际应用环境中，造成这种不匹配主要有三种因素：背景噪音、传输信道噪声和说话人的情感。噪声鲁棒技术的研究重点是如何减少背景噪声和传输信道噪声对种不匹配造成的影响[54]。

2.2.1 声学环境中的噪声和信噪比

人的语音从嘴说出开始到最终数字化所经历的一系列变化被定义为声学环境。在这个过程中主要有两种类型的干扰源：加性噪声（Additive noise）和信道噪声（Channel noise）。常见的加性噪声有：风扇转动时的噪声，开门关门的声音，其他说话人的声音等，这些声音都是我们日常生活中常见到的。信道噪声主要包括混响（Reverberation）、麦克风响应频率、A/D 转换电路的电流噪声、电话线路噪声、语音编码等。

2.2.1.1 加性噪声与信道噪声

加性噪声可以分为平稳（Stationary）噪声和非平稳（Non-stationary）噪声。平稳噪声在整个时间段内其功率谱比较稳定，例如电脑风扇和空调噪声。非平稳噪声其功率谱具有一定的统计特性，并随着时间的改变而变化，例如收音机、电视机以及其他说话人噪声。实际上，绝大部分加性噪声都是非平稳噪声，即使是来自电脑风扇、空调和汽车的噪声等，它们都不是完美的平稳噪声。

设带噪的加性语音信号为：

$$y[n] = s[n] + d[n] \tag{2.46}$$

其中 $s[n]$ 是纯净的语音信号，$d[n]$ 是背景噪声信号。

信道噪声也称为卷积噪声，语音信号被响应为 $h[n]$ 的信道噪声污染后可以表示为：

$$y[n] = s[n] * h[n] \tag{2.47}$$

在现实世界中，语音信号既受到加性噪声 $d[n]$ 的影响，也受到信道噪声 $h[n]$ 的影响，因此带噪的语音信号可表示为：

$$y[n] = s[n] * h[n] + d[n] \tag{2.48}$$

2.2.1.2 信噪比概念

为了度量加性噪声语音信号中噪声的“多少”或“含量”，引入了信噪比（Signal-to-Noise Ratio，SNR），信噪比定义为

$$SNR = 10\log_{10}\left(\sum_{i=1}^{L} s^2[n] \Big/ \sum_{i=1}^{L} d^2[n]\right) \tag{2.49}$$

式中 L 表示语音信号和噪声信号的总长度，即采样点个数。显然当信噪比越大时，信号中的噪声“含量”越少；当信噪比越小时，信号中的噪声“含量”越多。为了能控制语音信号中的噪声的含量，一般将式(2.46)改写为

$$y[n] = s[n] + \lambda \bullet d[n], \quad \lambda > 0 \tag{2.50}$$

式中λ是调控因子，可以调节带噪语音信号中噪声的“含量”，可根据式(2.49)得出其与信噪比的关系，即

$$SNR = 10\log_{10}\left(\sum_{i=1}^{L} s^2[n] \Big/ \sum_{i=1}^{L} (\lambda \bullet d[n])^2\right) \tag{2.51}$$

$$\lambda = \sqrt{\left(\sum_{i=1}^{L} s^2[n] \Big/ \sum_{i=1}^{L} d^2[n]\right) \bullet 10^{-SNR/10}} \tag{2.52}$$

因此，当给定纯净的语音信号、背景噪声信号以及信噪比，我们便可以根据式(2.50)和式(2.52)计算出带噪语音信号。

2.2.2 语音增强技术

信号空间的级降噪处理最常用的技术：语音增强。语音增强的主要目标是在接收端尽可能地从原始语音信号中提取纯净的语音信号。然而，由于干扰具有随机性，从带噪语音中提取完全纯净的语音几乎变得不可能。因此，在这种趋势下，语音增强的目的主要有两个：一是改善语音质量，消除背景噪音，使听者乐于接受，不感觉疲劳，这是一种主观度量，可用于人类听力感知；二是提高语音可懂度，这是一种客观度量，用于语音识别，如图 2.11 所示。然而，绝大多数情况下，这两个目的往往不能兼得。

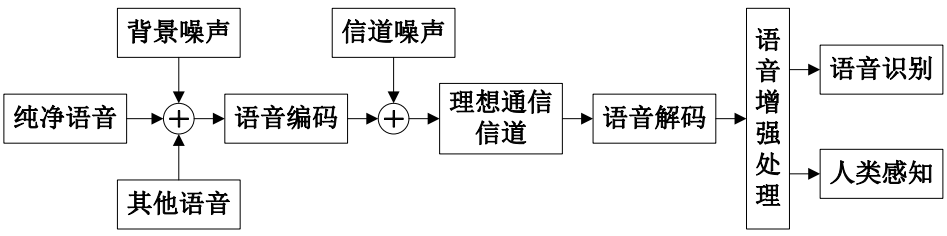

图 2.11 语音增强应用的一般流程

语音增强算法可以分为两类：单信道语音增强和双信道或多信道语音增强。在单信道语音增强应用中，仅有一个麦克风可用，因此必须假设在静音段到语音段之间的是背景噪声，进而获得背景噪声的统计特性。在多信道语音增强应用中，声音到达每个麦克风都有一个不同的时间差，也即一个信号是另一个信号的时延，多信道语音增强假中设主信道即包含语音信号又包含噪声信号，次信道中只包含噪声信号，然后利用主信道和次信道的共同作用一起去除噪声。

在本小节中，将主要介绍语音增强算法的一般流程和几种常用的单信道语音增强算法。

2.2.2.1 语音增强处理的一般流程

频域的语音增强算法基于短时分析技术[55]。输入的带噪语音信号首先需要分帧和加窗，然后进行 FFT 变换到频域，在频域上进行噪声的估计，增益函数的估计，进而获得纯净语音信号的估计频谱，并结合带噪语音的相位信息经傅里叶逆变换，最终得到增强后的语音信号，如图 2.12 所示。

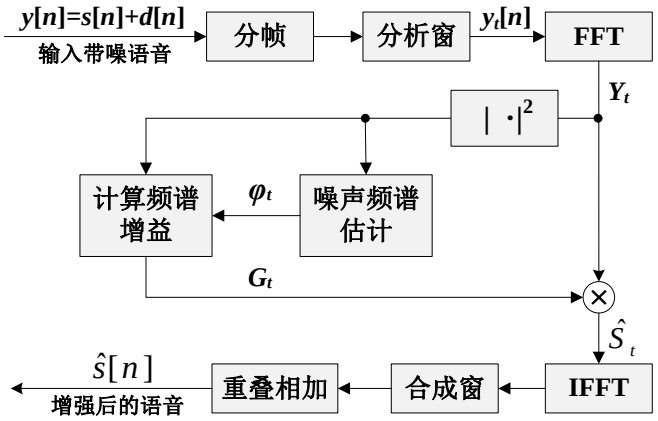

图 2.12 语音增强处理流程图

在语音增强中，一般假设噪声是加性的且是局部平稳的。局部平稳是指一段带噪语音中的噪声，具有和语音开始前那段噪声相同的统计特性，且在整个语音段中保持不变。也就是说，可以根据语音开始前那段噪声来估计语音中所叠加的噪声统计特性。另一个假设是假设噪声与语音统计独立或不相关，这样可以忽略交叉项。

设加窗分帧后的第 t 帧带噪语音 $y_t[n]$ 表示为

$$y_t[n] = s_t[n] + d_t[n], \quad 0 \le n \le N-1 \tag{2.53}$$

其中 $s_t[n]$ 和 $d_t[n]$ 分别是第 t 帧纯净语音和噪声， N 是语音帧长。对 $y_t[n]$ 进行短时傅里叶变换得

$$Y_t(w_k) = S_t(w_k) + D_t(w_k) \tag{2.54}$$

式中 $w_k = 2\pi k/N$ （ $k = 0,1,\ldots,N-1$）。

假如这时已经求得噪声的抑制函数（Suppression function）$G_t(w_k)$，也称增益函数（Gain function），便可以通过下式估计得出纯净语音的估计频谱

$$\hat{S}_t(w_k) = G_t(w_k) \bullet Y_t(w_k) \tag{2.55}$$

上式中，因为语音幅度谱大于 0 且不能超过带噪语音的幅度值，所以增益函数应当满足 $0 \le G_t(w_k) \le 1$。当获得频谱后，就可以通过傅里叶逆变换获得增强后的语音信号。

因此，以上的分析可以看出语音增强的关键便是如何求得增益函数。

2.2.2.2 谱减法及其改进算法

谱减法是语音增强中最早使用的一种方法，这种方法基本思想是：直接从带噪语音信号的频谱中减去噪声的平均频谱。对式(2.54)平方求功率谱得

$$\left|Y_t(w_k)\right|^2 = \left|S_t(w_k)\right|^2 + \left|D_t(w_k)\right|^2 + S_t(w_k) \bullet D_t^*(w_k) + S_t^*(w_k) \bullet D_t(w_k) \tag{2.56}$$

如假设语音信号与噪声信号都是 0 均值，且它们之间统计不相关，也就是说交叉项为 0，则式(2.56)可近似为：

$$\left|Y_t(w_k)\right|^2 \approx \left|S_t(w_k)\right|^2 + \left|D_t(w_k)\right|^2 \tag{2.57}$$

在式(2.57)中，尽管 $\left|D_t(w_k)\right|^2$ 未知，但我们可以通过求带噪语音前若干帧的平均值来估计（假设带噪语音段的前若干帧都是噪声），即

$$\left|\hat{D}(w_k)\right|^2 = \frac{1}{M}\sum_{t=0}^{M-1}\left|Y_t(w_k)\right|^2 \tag{2.58}$$

因此可根据式(2.57)和式(2.58)得到一个比较直观的谱减法，即

$$\left|\hat{S}_t(w_k)\right|^2 = \left|Y_t(w_k)\right|^2 - \left|\hat{D}(w_k)\right|^2 = \left|Y_t(w_k)\right|^2\left(1 - \frac{\left|\hat{D}(w_k)\right|^2}{\left|Y_t(w_k)\right|^2}\right) \tag{2.59}$$

若定义与频率相关的后验信噪比（Posteriori SNR）$\varphi_t(w_k)$ 为

$$\varphi_t(w_k) = \frac{\left|Y_t(w_k)\right|^2}{\left|\hat{D}(w_k)\right|^2} \tag{2.60}$$

需要注意的是这里的后验信噪比去前面定义的信噪比是有区别的，后验信噪比跟时间和频率相关，是一个瞬时量。于是便可以根据式(2.55)、式(2.59)和式(2.60)求得谱减法的增益函数

$$G_t(w_k)=\sqrt{1-\frac{1}{\varphi_t(w_k)}} \tag{2.61}$$

由于语音的功率谱非负，因此后验信噪比必须满足$\varphi_t(w_k)\geq 1$。但由于噪声的随意性并且估计时也会存在偏差，所以在实际应用中这一点很难保证，为此有些文献[23]便对其设定一个下限，则式(2.61)可改写为

$$G_t(w_k)=\sqrt{\max\left(1-\frac{1}{\varphi_t(w_k)},a\right)} \tag{2.62}$$

式中a为一常数，称为衰减因子。

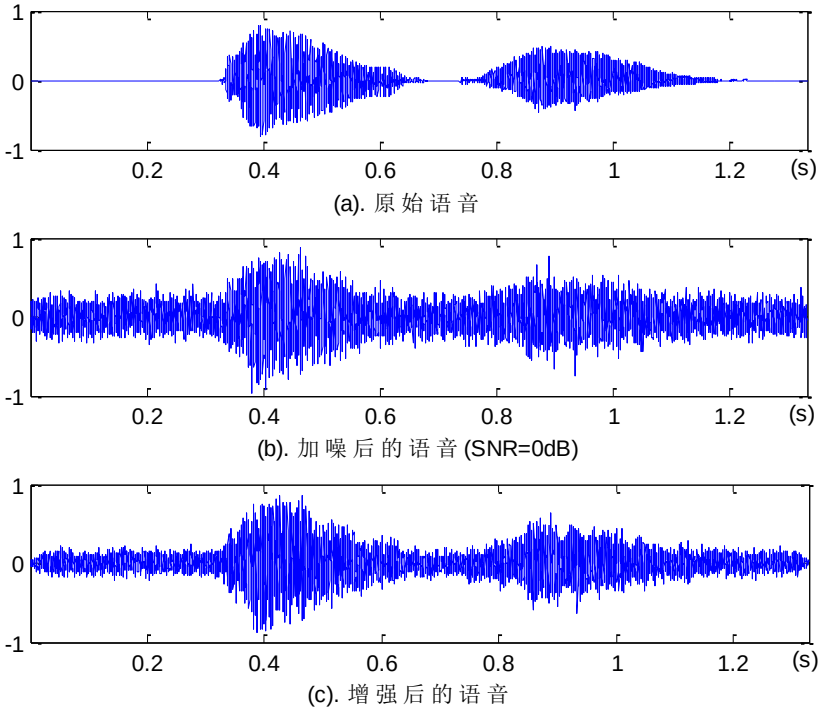

图 2.13 谱减法效果图

使用谱减法能很好的抑制带噪语音中的噪声，但却残留下了另一种非常刺耳的噪声，称为音乐噪声[24]。造成该现象的主要原因是式(2.58)和式(2.61)估计噪声的能力很弱，噪声总是在不断的变化或噪声在较短的时间段内平稳。为了随时跟踪噪声的变化，需要一个实时的 VAD 程序。当 VAD 检测出第t帧是非语音帧时，利用下式对噪声进行重新估计：

$$\left|\hat{D}_t(w_k)\right|^2=\tau\bullet\left|\hat{D}_{t-1}(w_k)\right|^2+(1-\tau)\bullet\left|Y_t(w_k)^2\right| \tag{2.63}$$

造成音乐噪声的另一个原因是后验信噪比$\varphi_t(w_k)$的估计在各个频率分量上进行，并没有考虑各个频率分量之间的相关性。因此有人提出在频率分量间进行平滑，这种方法可以在不引起信号失真的情况下能很好的抑制噪声，并能提升听

觉效果。同样，在时间上进行平滑也可以减少失真，即

$$\varphi_t(w_k)=\gamma\bullet\varphi_{t-1}(w_k)+(1-\gamma)\frac{|Y_t(w_k)|^2}{|\hat{D}(w_k)|^2},\quad 0<\gamma<1 \tag{2.64}$$

式中γ是平滑因子，谱减法的效果图 2.13 所示。

对于谱减法还有其他许多改进方法[25-27]，如式(2.62)中的衰减因子a也可以变成频率的函数，当我们想使某个频率分量得到更多的衰减，这将变得非常有用；式(2.64)中只对相邻的两帧进行了平滑，我们也可以对前面多帧进行平滑；对于式(2.62)的增益函数还可以变成任意阶，即

$$G_t(w_k)=\left(\max\left(1-\frac{1}{\varphi_t^{\alpha/2}(w_k)},a\right)\right)^{1/\alpha} \tag{2.65}$$

上式中，若$\alpha=2$便对应着功率谱减法，若$\alpha=1$便是幅度谱减法。

2.2.2.3 维纳滤波及其改进算法

维纳滤波器[56][57]的主要目的是滤除带噪语音信号中的噪声，它是一种基于统计模型的估计方法。

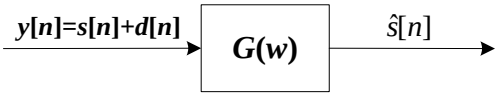

图 2.14 维纳滤波器输入与输出

如图 2.14 所示，设维纳滤波器是一单位脉冲响应为$g[n]$的线性滤波器。输入带噪语音信号$y[n]=s[n]+d[n]$，则线性系统的输出为

$$\hat{s}[n]=y[n]*g[n]=\sum_{m=-\infty}^{+\infty}g[m]y[n-m] \tag{2.66}$$

式中$\hat{s}[n]$是估计得出的纯净语音信号，它和我们期望得到的纯净语音信号$s[n]$会存在误差

$$e[n]=s[n]-\hat{s}[n] \tag{2.67}$$

显然$e[n]$是随机变量，维纳滤波器的基本思想就是使误差平方的期望最小，即

$$E\left(e^2[n]\right)=E\left((s[n]-\hat{s}[n])^2\right) \tag{2.68}$$

对式(2.66)和式(2.68)使用最小均方误差准则（MMSE）求解，并变换到频域，最终的得到

$$G_t(w_k)=\frac{|S_t(w_k)|^2}{|D_t(w_k)|^2+|S_t(w_k)|^2} \tag{2.69}$$

从上式可以看出，语音的功率谱$|S_t(w_k)|^2$是一个未知量，这便形成了一个先有鸡还是先有蛋的问题。为了解决这一问题，我们需要从另一角度入手，定义先验信噪比（Priori SNR）为

$$\xi_t(w_k)=\frac{|S_t(w_k)|^2}{|D_t(w_k)|^2} \tag{2.70}$$

将式(2.70)代入式(2.69)得：

$$G_t(w_k)=\frac{\xi_t(w_k)}{1+\xi_t(w_k)} \tag{2.71}$$

从式(2.60)和式(2.70)可以看出，先验信噪比和后验信噪比满足

$$\xi_t(w_k)=\varphi_t(w_k)-1 \tag{2.72}$$

从上式表明：我们可以由后验信噪比$\varphi_t(w_k)$得到先验信噪比$\xi_t(w_k)$，进而求得增益函数$G_t(w_k)$。

另一方面，式(2.66)变换到频域得

$$\hat{S}_t(w_k)=G_t(w_k)\bullet Y_t(w_k) \tag{2.73}$$

将其带入式(2.70)

$$\xi_t(w_k)=\frac{|S_t(w_k)|^2}{|D_t(w_k)|^2}\approx\frac{|\hat{S}_t(w_k)|^2}{|D_t(w_k)|^2}=\frac{|G_t(w_k)\bullet Y_t(w_k)|^2}{|D_t(w_k)|^2}=|G_t(w_k)|^2\varphi_t(w_k) \tag{2.74}$$

到这里，我们可以从式(2.72)和式(2.74)看出，通过后验信噪比$\varphi_t(w_k)$可以有两种方式求得先验信噪比$\xi_t(w_k)$。在实际应用中，通常对两者结合使用，即

$$\xi_t(w_k)=\alpha\bullet G_{t-1}^2(w_k)\varphi_{t-1}(w_k)+(1-\alpha)\bullet\max(\varphi_t(w_k)-1,0),\quad 0\le\alpha\le 1 \tag{2.75}$$

上式中等号右边的第一项表示前一帧的先验信噪比的信息，第二项表示当前帧的先验信噪比的信息。式中α是平衡因子，用于调节两种方法估计先验信噪比所占的比重，α越大，残留的音乐噪声越少，但语音的失真越大。维纳滤波语音增强的效果如图 2.15 所示（$\alpha=0.98$）。

维纳滤波也有很多改进算法，其中大多数都是前面谱减法中所提及到的改进，这里将不再详述。

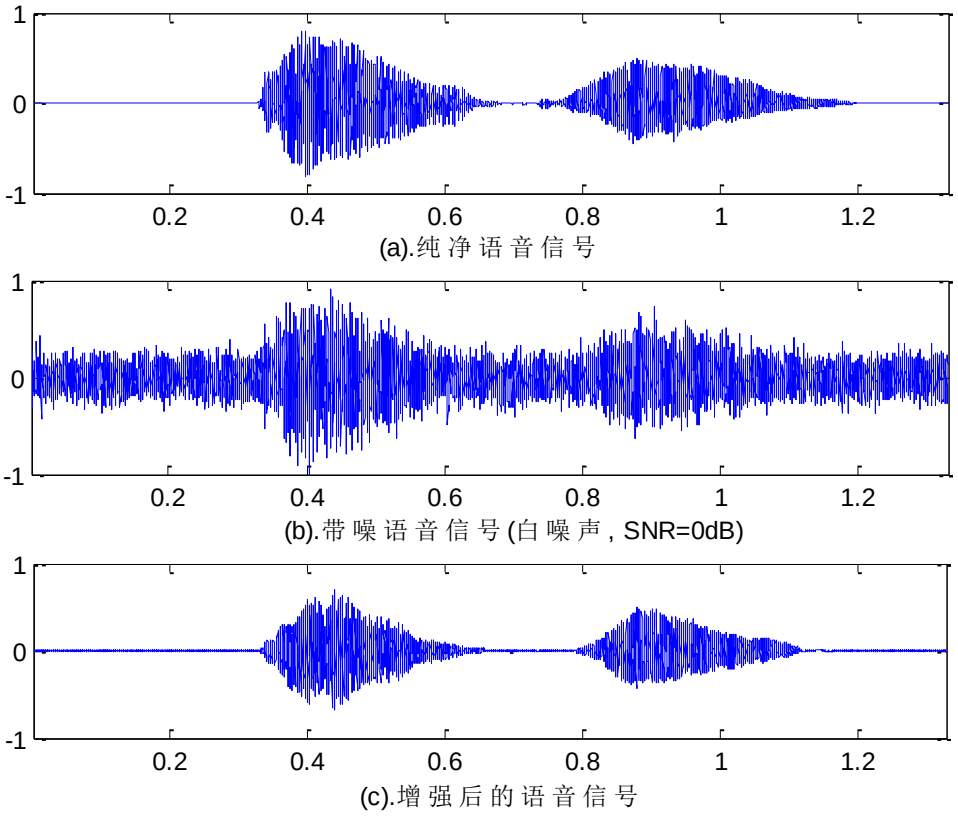

图 2.15 维纳滤波语音增强

2.2.3 特征空间噪声鲁棒技术

特征空间噪声鲁棒技术的主要目的是获得不依赖环境、能快速适应噪声的变化，且具有简单分布（单高斯模型）的统计参数。

在本小节中主要讨论特征参数的几种线性和非线性变换技术[31][32]，这些技术简单、易实现并且功能强大，能很好的处理由各种噪声引起的训练环境和识别环境不匹配，特别地是对卷积噪声的处理效果更为明显，进而提升了语音识别系统的鲁棒性。

2.2.3.1 倒谱特征变换

设语音信号经前端处理后的输出特征参数序列为 $\mathbf{O}=\{\mathbf{o}_1,\mathbf{o}_2,\ldots,\mathbf{o}_t,\ldots,\mathbf{o}_T\}$，其均值和方差为

$$\boldsymbol{\mu}=\frac{1}{T}\sum_{t=1}^{T}\mathbf{o}_t\ ,\quad \boldsymbol{\sigma}^2=\frac{1}{T}\sum_{t=1}^{T}(\mathbf{o}_t-\boldsymbol{\mu})^2 \tag{2.76}$$

则倒谱均值减（Cepstral Mean Subtraction，CMS）或倒谱均值归一（Cepstral Mean Normalization，CMN）定义为

$$\bar{\mathbf{o}}_t=\mathbf{o}_t-\boldsymbol{\mu} \tag{2.77}$$

由信道产生的噪声一般是卷积噪声，前面曾讨论过，时域的卷积变换到倒谱域便是求和，因此对于倒谱特征向量$\mathbf{o}_t$可以表示成纯净语音信号的倒谱$\mathbf{s}_t$与信道倒谱$\mathbf{h}_t$的和，即

$$\mathbf{o}_t = \mathbf{s}_t + \mathbf{h}_t \tag{2.78}$$

对于同一麦克风而言，其统计特性几乎固定，即$\mathbf{h}_t \approx \mathbf{h}$，当经 CMS 操作时有

$$\bar{\mathbf{o}}_t = \mathbf{o}_t - \boldsymbol{\mu} \approx \mathbf{s}_t + \mathbf{h} - \frac{1}{T}\sum_{i=1}^{T}(\mathbf{s}_i + \mathbf{h}) = \mathbf{s}_t - \frac{1}{T}\sum_{i=1}^{T}\mathbf{s}_i \tag{2.79}$$

从上式可以看出当经 CMS 的线性操作后可以移除卷积噪声的干扰。同样，对于加性噪声而言，在信噪比较大时，也即语音信号中的混杂的噪声少时，CMS 能起到很好的效果，具体的分析可以阅读文献[58]。

另外一种常用的特征变换方法是倒谱均值方差归一（Cepstral Mean and Variance Normalization，CMVN），定义为

$$\hat{\mathbf{o}}_t = \frac{\mathbf{o}_t - \boldsymbol{\mu}}{\boldsymbol{\sigma}} \tag{2.80}$$

从上式可以看出 CMVN 不仅具备了 CMS 的线性特性，还有自己的非线性部分。不同的语音段可能有着不同的方差，当除以方差时可以看着自增益控制，因此 CMVN 不仅移除了由噪声造成的均值偏移，同时也压缩了特征空间的取值范围。另一方面，从统计角度来说，特征序列经式(2.80)处理后，将更加接近于高斯分布。

在使用 CMS 和 CMVN 需要注意下列一些问题：

1. 训练和测试要么同时使用 CMS 或 CMVN，要么都不使用。从式(2.79)和式(2.80)可以看出，当使用后 CMS 或 CMVN，原先的特征空间将变换到另外一个特征空间，因此我们必须在相同的特征空间进行训练和测试，否则会再一次造成训练和测试不匹配；

2. 尽管在使用 CMS 和 CMVN 后，系统的识别率在信噪比较低时有很大的提高，但在使用 CMS 和 CMVN 进行特征空间变换时，会对原先的特征产生一些失真，使得在较高信噪比时，系统的识别率会有轻微地下降；

3. CMS 和 CMVN 共同面临的一个问题是它们不能区分有效语音段和静音段。因为它们在计算均值和方差时，是对整段语音进行求取。若一段语音中的静音段较多时，必然会对均值和方差造成影响，从而影响识别效果。

4. 若 T 非常大，即特征序列非常长，则均值和方差必须在所有特征矢量都准备好后才能计算，这样不利于实时系统的实现。

2.2.3.2 分段倒谱特征变换

为了克服整段语音求取均值和方差的问题，文献[59]中提出了分段特征向量归一化（Segmental feature vector Normalization）方法，如图 2.16 所示。

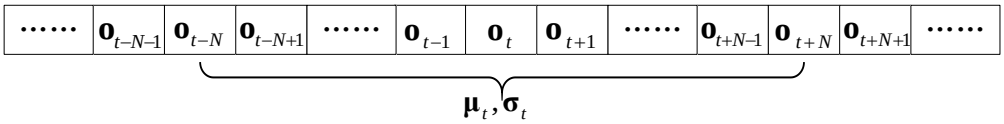

图 2.16 分块倒谱特征参数归一

此时，第 t 帧的均值和方差的计算可写成

$$\boldsymbol{\mu}_t = \frac{1}{2N+1}\sum_{i=t-N}^{t+N}\mathbf{o}_i \tag{2.81}$$

$$\boldsymbol{\sigma}_t^2 = \frac{1}{2N+1}\sum_{i=t-N}^{t+N}\left(\mathbf{o}_i - \boldsymbol{\mu}_t\right)^2 \tag{2.82}$$

于是 CMS 和 CMVN 改写成：

$$\bar{\mathbf{o}}_t = \mathbf{o}_t - \boldsymbol{\mu}_t \tag{2.83}$$

$$\hat{\mathbf{o}}_t = \frac{\mathbf{o}_t - \boldsymbol{\mu}_t}{\boldsymbol{\sigma}_t} \tag{2.84}$$

从式(2.81)和式(2.82)可以看出，均值和方差的计算是在距中心点半径为 N 的局部区域进行。因此，分段 CMS 和 CMVN 不仅有利于实时系统的实现，而且还能有效地避免语音段中静音段过多的情况，因为即使有太多的静音段也只是影响局部区域，而不是影响整段语音。在式(2.81)和式(2.82)的中，计算特征序列头尾若干帧的均值和方差时，会有边界溢出。为解决该问题，有两种方案：一种是对于溢出的值全用 0 代替；另一种是用第一帧或最后一帧代替。

在使用该分块算法的过程中，我们还需要注意的是 N 的取值，不能太小，也不能太大。太小则不能反映出其统计特性，太大就会出现前面所提到的问题。一般而言，N 的取值区间是 20~40。在实际的应用中一般都采用分段的 CMS 和 CMVN，本文以及后续的实验也是基于分段的倒谱特征变换。

图 2.17 中展示了纯净语音和带噪语音的 MFCC、CMS 和 CMVN 的第二维特征随时间变化的曲线（N=30）。从（a）和（b）可以看出，当纯净语音收到噪声干扰时，其取值范围和中心都会发生变化。当经过 CMS 处理后，这时纯净语音和带噪语音的输出特征均值都为 0，但取值范围却不相同，如在（c）中纯净语音的输出特征，取值范围大约在-20~20 之间，而在（d）中带噪语音的输出特征取值范围大约在-10~10 之间。在（e）和（f）中是 CMVN 的输出特征，可以

看出它们的值域范围大致相同。

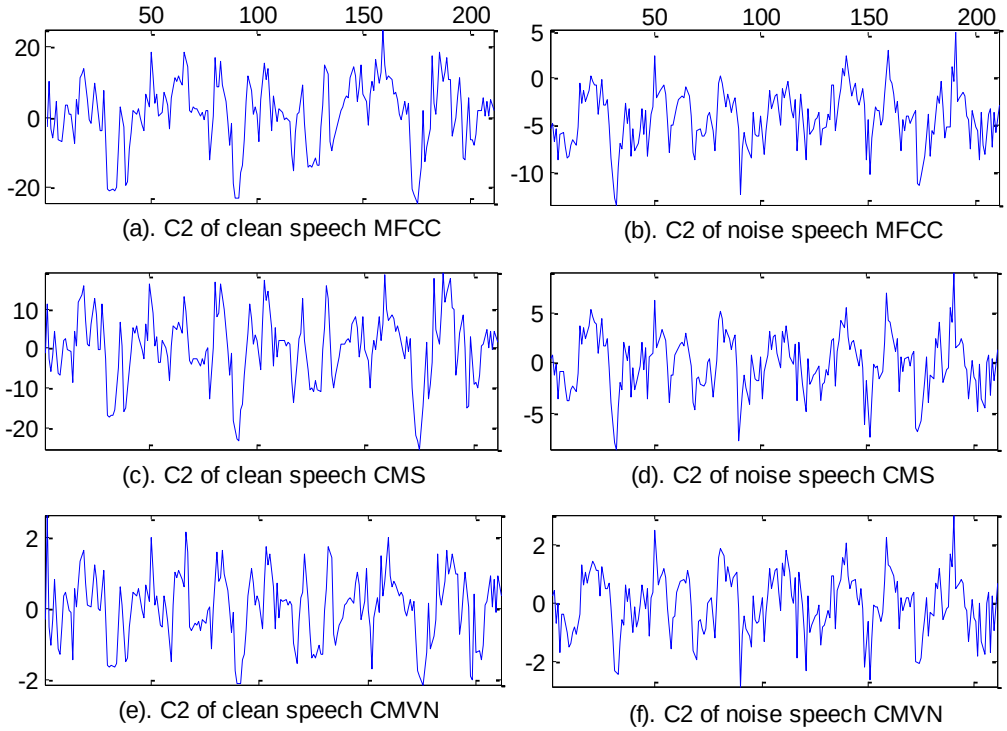

图 2.17 纯净语音和带噪语音（SNR=0）的各种特征对比（一）

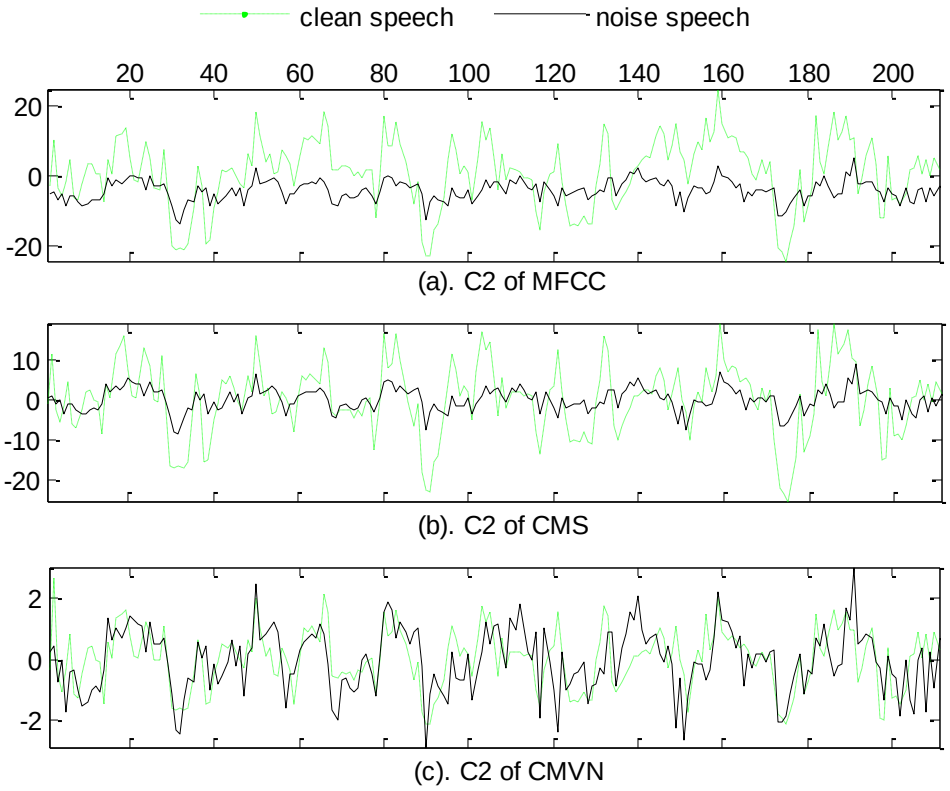

图 2.18 纯净语音和带噪语音（SNR=0）的各种特征对比（二）

若将图 2.17 中的纯净语音和带噪语音的三种特分别征绘制在一起，如图 2.18 所示，可以看出纯净语音和带噪语音的特征经 CMVN 处理后最为相似。因此可以直观地推断出 CMVN 比 CMS、MFCC 具有更好的鲁棒性。这一观点将在本章的实验得到证实。

在分段倒谱特征参数变换算法中，对于每一帧都需要以其为中心，各计算一次均值和方差，均值和方差变成了与时间相关的量。因此，这也毫无疑问地增加了计算量。

2.3 小结

语音识别技术及其噪声鲁棒性技术发展到现在已有半个多世纪，这期间无数科学家、学者都投入这方面的研究，也涌现出了各种不同的技术，浩如烟海。本章从实际应用的嵌入式平台出发，主要围绕常用的语音识别技术和噪声鲁棒性技术进行介绍，从语音信号的采集量化，预处理，语音增强，特征提取，鲁棒性特征处理，声学建模，到最终的模式匹配等，有了这些基础知识便可以构建一个简单的识别系统。

第3章 噪声鲁棒语音识别仿真系统搭建

前面章节中介绍了语音识别技术及其噪声鲁棒性技术，本章的主要内容是根据前面介绍的技术搭建一个噪声鲁棒语音识别实验平台，用于噪声鲁棒性算法的验证与测试，以及后期用于指导嵌入式语音识别系统的实现。

3.1 实验数据准备

3.1.1 语音数据库

本文实验所用的语音数据由 56 人在实验室环境下录制的，其中男性 31 人，女性 25 人，每人对 10 个中文词发音，每个词 2~3 个字，每个词发音 5 次，共 56×10×5=2800 个语音段。语音数据的采样率为 16kHz，16bit 量化，单信道，wav 格式，PCM 编码。

3.1.2 噪声数据库

实验采用的噪声数据库是由 Institute for Perception-TNO 和 Speech Research Unit 在 1992 年联合录制的标准噪声库 Noisex-92[60][61]。数据库中包含许多不同类型的噪声采样数据，本文主要使用了其中的白噪声（White noise）、粉色噪声（Pink noise）、汽车行驶过程中的车内噪声（Volvo 340 noise）、多人谈话噪声（Babble noise）。白噪声是最常见的宽带噪声之一，它的每个频带内的能量为常数，且基本恒定，很难用高通或低通滤波器进行处理；粉色噪声在给定频率范围内（不包含直流成分），随着频率的增加，其功率密度每倍频下降 3dB（密度与频率成反比）；Volvo 340 噪声在雨天泊油路上，以 120km/h 行驶的汽车车内录制；多人谈话噪声是在 100 人的餐厅录制，餐厅的半径约 2 米，因此很难听清某一个人的声音，Babble 噪声和人类语音相关性较大。这些噪声采样率都是 19.98kHz，16bit 量化。在前面介绍的语料库中，语音数据的采样频率都是 16kHz，因此需要对这些噪声数据进行降采样处理，降采样的工具使用 Cool Edit Pro 2.0[62]。

3.2 实验仿真系统搭建

为了便于算法测试和实验，以及后期引导嵌入式语音识别系统的完成，本文

根据前面所介绍的语音识别技术，构建了一个基于 matlab 的 NLP+GMM 语音识别实验平台。该实验平台包括数据读入、端点检测、特征提取、模型训练和模式识别等功能。该系统可以进行特定人和非特定人的语音识别实验。后期为了进行噪声鲁棒性的测试，又为该平台添加了语音信号加噪、语音增强、特征变换等功能。图 3.1 中显示了整个仿真系统的结构图。

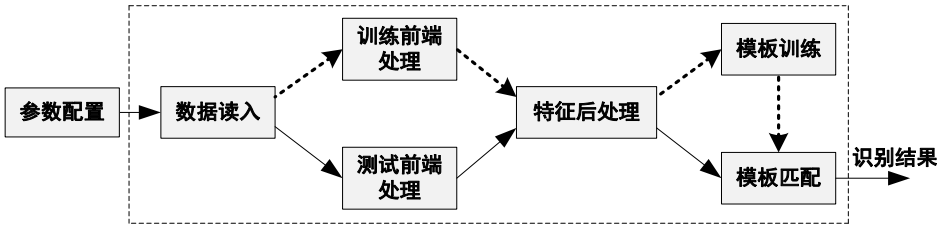

图 3.1 仿真系统结构图

3.2.1 系统参数配置模块

语音识别系统中同一功能的模块存在许多不同的算法，如在端点检测中，可以根据音量进行端点检测，也可根据音量和过零率进行端点检测。还有许多不同的配置参数，如语音的采样频率、每个词划分的状态数、高斯混合密度函数的个数等。不同算法和不同参数配置项设置，可取得不同的实验结果。为了对这些算法和参数进行灵活的管理，设计了系统的参数配置模块。

图 3.2 中是基于 matlab 语音实验平台的一些重要参数配置，包括语音数据的位置、文件标识、语音数据的采样率、端点检测算法、特征提取配置等。

```
function vcrParam=vcrParamSet
vcrParam.waveDir='E:\SpeechCorpora\trainSet';        %训练语音文件夹
vcrParam.waveTypeIdx=1;                %文件名字标识 wav 文件类型的索引
vcrParam.fs = 8000    %  语音文件采样率
%  端点检测参数配置  0：基于音量的端点检测，1：音量和过零率
vcrParam.useEpd=1;
[vcrParam.mfccParam, order] = mfccExtParamSet(vcrParam.fs, 1); %MFCC 特征参数配置
vcrParam.cmsParam = CMSParamSet;    %倒谱均值减参数配置
vcrParam.cmvnParam = CMVNParamSet;    %倒谱均值方差归一参数配置
vcrParam.nlpgmmParam = nlpgmmParamSet(order);    %模型初始化参数配置
vcrParam.nlpgmmParam.stateNum = 4;            %  状态数
%  每个状态的高斯混合数
vcrParam.nlpgmmParam.gussNum = 7*ones(vcrParam.nlpgmmParam.stateNum,1);
vcrParam.feaType = 1; %特征参数类型选项，  1:mfcc, 2: cms, 3: cmvn, 5: stcmvn
vcrParam.spEnOpt = 0;    %语音增强算法选项，0：  不使用语音增强，1：SS，2：WF
%  测试阶段端点检测选项，0：纯净语音结果，1：增强后的结果，2：带噪语音的结果
```

```
vcrParam.vadOpt = 0;
vcrParam.testOpt = 0;   %测试阶段输入信号选择，0：原始带噪语音，1：增强后的语音
vcrParam.noiseParam=NoiseParamSet(vcrParam.fs); %噪声数据库选择
……
```

图 3.2 实验平台的部分参数配置清单

3.2.2 数据读入模块

语音数据存储在磁盘上，为了对这些数据进行处理，需要按照一定的数据结构将它们读入到内存中。图 3.3 中展示了语音数据读入模块的部分代码。最后将数据分成训练集和测试集，并进行存储。

```
function [speakerData speakerNum]=readSpeakerData(vcrParam)
dirName = vcrParam.waveDir;
classIdx = vcrParam.waveTypeIdx;
speakerData=dir(dirName);     %获得说话人的目录
speakerNum=length(speakerData);   %录音人数量
for i=1:speakerNum
    waveFiles = dir([dirName, speakerData(i).name, '/*.wav']);
    waveFiles=waveFiles(digitIndex);
    waveNum=length(waveFiles); %语音文件数量
    for j=1:waveNum
        [y, fs, nbits] = wavread(fullPath);%wavReadInt(fullPath); %读入语音数据
        speakerData(i).sentence(j).y=y;
        speakerData(i).sentence(j).fs=fs;
        speakerData(i).sentence(j).class=waveFiles(j).name(classIdx);
        %  端点检测
        if vcrParam.useEpd==1         %  音量
            speakerData(i).sentence(j).endPoint= epdByVol(y, fs, nbits);
        elseif vcrParam.useEpd==2    %  音量与过零率
            speakerData(i).sentence(j).endPoint= epdByVolZero(y, fs, nbits);
        else       %  不作端点检测
            speakerData(i).sentence(j).endPoint = [1, length(y)];
        end   % end if
        ……
    end   % end for j
end   % end for i
……
save speakerData_Train speakerData(1:35)   %保存数据用于训练
save speakerData_Test speakerData(36:end)   %保存数据用于测试
```

图 3.3 数据读入部分代码清单

3.2.3 前端处理模块

语音识别的前端处理指从输入语音信号到输出最终特征序列的这一过程。平台中直接使用实验室录制的纯净语音数据进行特征提取。在测试中，首先需要对测试数据加上不同信噪比的不同噪声，然后在进行一些鲁棒性技术处理，最后输出特征序列。图 3.4 中显示了测试阶段中前端处理的加噪、语音增强和 MFCC 特征提取等的部分代码清单。

```
function speakerData=spkrData_EN_VAD_MFCC(speakerData, vcrParam)
% 语音增强，增强后的语音信号存放在<speakerData.sentence.es>中
speakerData=spkrDataEnhance(speakerData, vcrParam);
if vcrParam.vadOpt==1   % 增强后的语音用作端点检测
    speakerData = VAD_On_EnhancedSpeech(speakerData);
elseif vcrParam.vadOpt==2    % 原始带噪语音用作端点检测
    speakerData = VAD_On_NoiseSpeech(speakerData);
end
if vcrParam.testOpt   % 增强后的语音用作特征提取
    speakerData = MFCC_On_EnhancedSpeech(speakerData, vcrParam);
else   % 原始带噪语音用作特征提取
    speakerData = MFCC_On_NoiseSpeech(speakerData, vcrParam);
end
% 获得特征参数后释放内存
for i=1:length(speakerData)
    for j=1:length(speakerData(i).sentence)
        speakerData(i).sentence(j).ns=[];
        speakerData(i).sentence(j).es=[];
    end
end
```

图 3.4 测试阶段前端处理：加噪、语音增强、特征提取

在图 3.5 中显示了测试阶段前端处理中的特征变换和非线性分段部分代码清单。特征变化包括 CMS，CMVN 等算法。经过这一步后便可以输出最终的声学特征序列和非线性分段算法的分段结果。

```
function speakerData=feaProcssing(speakerData, vcrParam)
for i=1:length(speakerData)
    for j=1:length(speakerData(i).sentence)
        fea = speakerData(i).sentence(j).fea;   % 获取 MFCC 特征
        fea = getFeature(fea, vcrParam);        % 特征变换处理
        speakerData(i).sentence(j).fea = fea;
        % 对输出的特征进行非线性分段
```

```
            speakerData(i).sentence(j).partition=NonLinearPartition(fea, vcrParam.nlpgmmParam);
            clear fea   % 释放内存
        end
end

function fea=getFeature(fea, vcrParam)   %特征处理
switch vcrParam.feaType
        case 2   % 倒谱均值减
            fea=mfcc2cms(fea, vcrParam.cmnParam);
        case 3 % 倒谱均值方差归一
            fea=mfcc2cmvn(fea, vcrParam.cmnParam);
        case 5   % Statistic threshold CMVN
            fea=fea2stcmvn(fea, vcrParam.stcmvnParam);
        ……
        otherwise
            %disp('Warning: The feature type is not exist, using the mfcc to substitute');
end
```

图 3.5 测试阶段前端处理：特征变换和非线性分段

3.2.4 模型训练

在训练过程中当获得声学特征和分段信息后，可以采用 NLP+GMM 对每个识别单元进行建模，调用函数 nlpgmmTrain 可训练出识别单元的模型。测试时，将输入语音经前端处理后，调用函数 nlpgmmRecognition 进行测试。图 3.6 中展示了模型训练和训练数据集内测试部分代码的清单。

```
function goModelTrain(vcrParam)
load speakerData_Train;   % 加载训练数据

% 模型训练
trainData=[speakerData_Train.sentence];   % 获得数据集
gmmTrainParam=gmmTrainParamSet;      % 高斯混合模型参数初始化
nlpgmmModel = nlpgmmTrain(trainData, vcrParam.nlpgmmParam, 1, gmmTrainParam);

% 集内测试
trainData=[speakerData_Train.sentence];
[recogRate1, confusionTable1] =
    nlpgmmRecognition(trainData, nlpgmmModel, vcrParam.nlpgmmParam);
……
save nlpgmmModel nlpgmmModel   %保存模板
```

图 3.6 模型训练与训练数据集内测试模块部分代码清单

3.2.5 噪声鲁棒性测试模块

为了进行噪声鲁棒性算法的验证与测试，需要在前端处理的预处理过程中给测试数据加上不同信噪比、不同类型的背景噪声。然后再分别进行测试，图 3.7 中给出了噪声鲁棒性测试模块的部分代码。

```
function noiseParam=goModelTest(vcrParam)
noiseParam=vcrParam.noiseParam;
for i=1:length(noiseParam)
    for j=1:length(noiseParam(i).SNRs)
        % 不同信噪比和不同背景噪声下的鲁棒性测试
        [noiseParam(i).SNRs(j).recogRate, noiseParam(i).SNRs(j).confusionTable] = …
            modelTestInNoise(vcrParam, noiseParam(i).SNRs(j).SNR, noiseParam(i).signal);
    end
end
% 根据信噪比和背景噪声进行鲁棒性测试
function [recogRate confusionTable]=modelTestInNoise(vcrParam, snr, noise)
load speakerData_Test;     % 加载测试数据
load nlpgmmModel;          % 加载模型
speakerData_Test=addNoise(speakerData_Test, snr, noise); % 测试语音加噪
% 测试语音前端处理，获得 MFCC 特征
speakerData_Test=spkrData_EN_VAD_MFCC(speakerData_Test, vcrParam);
% 测试语音前端处理，进一步对特征进行处理，如特征变换等
speakerData_Test=feaProcesing(speakerData_Test, vcrParam, 1);
% 外部测试
testData=[speakerData_Test.sentence];
recogRate = nlpgmmRecognition(testData, nlpgmmModel, vcrParam.nlpgmmParam);
```

图 3.7 噪声鲁棒性测试部分代码清单

3.3 语音增强的噪声鲁棒性实验

前一章中介绍了信号级降噪技术：语音增强。语音增强的两个主要目的是改善语音的质量和提升语音信号的可理解性，前者用于改善人的视听效果，而后者主要用于语音识别。本小节中，我们将根据前面已搭建好的 matlab 语音识别实验平台，进行语音增强算法实验。

3.3.1 系统参数设置

系统的预加重系统是 0.95，语音帧长 16ms，帧移为 0ms，使用汉明窗。声

学特征使用静态 MFCC+对数能量和它们的 1 阶动态特征，MFCC 静态特征输出维数是 12，因此最终特征参数的输出维数是 26 维。

在模型训练过程中，从实验数据库中随机挑选 35 人的语音数据用于训练，训练语音直接使用实验室环境下的纯净语音，不加任何噪声。训练时，每个词用非线性分段算法划分为 N=4 个状态，每个状态采用 GMM 建模，混合密度 M=7。剩下 56-35=21 人的用于测试，测试的语音在测试前都需要加上不同信噪比（SNR=-5，0，5，15，20dB）的不同噪声。在加噪后，带噪语音需要经过语音增强算法处理，即前面讨论的谱减法（SS）和维纳滤波（WF），然后将增强后的语音用于端点检测，特征提取，模式识别等。在整过实验过程中，训练语音不用在测试中。

3.3.2 实验结果与分析

实验结果如图 3.8 所示，实验中基线系统采用 MFCC 参数，“SS+MFCC”表示谱减法与 MFCC 结合，“WF+MFCC”表示维纳滤波与 MFCC 相结合。图中分别显示了这三种方法在白噪声、粉噪声、汽车车内噪声以及人群噪声中不同信噪比下的实验结果。

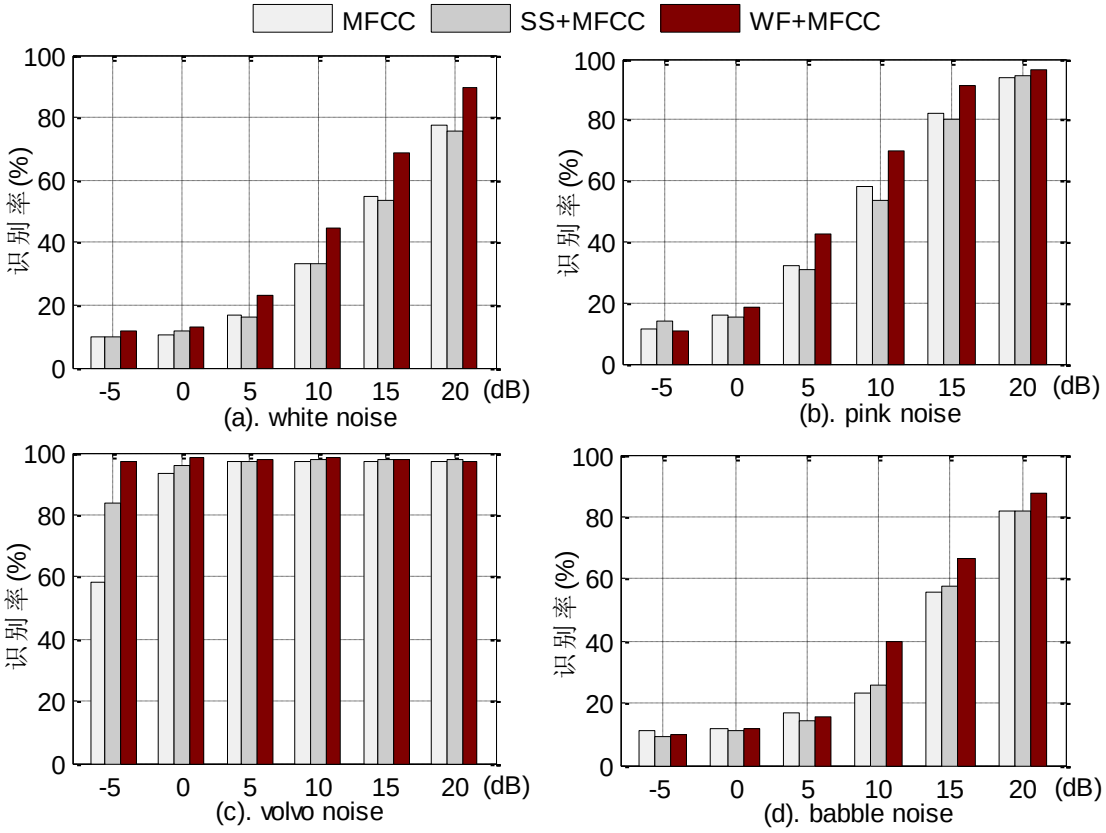

图 3.8 语音增强后的识别率

从图中可以看出三种方法在白噪声、粉噪声和人群噪声中随着信噪比的降低，识别性能大大的下降，对于较为平稳的汽车车内噪声，三种方法在较高信噪比时相差不大。另外从图中还可以看出：SS+MFCC 的效果有时比直接使用 MFCC 参数还要差，这是因为谱减法在去掉噪声时会残留下许多音乐噪声。WF+MFCC 的整体性能要好于其他两种。在 SNR=-5dB 时，SS+MFCC 和 WF+MFCC 在各种噪声情况下的平均识别率相对于 MFCC 分别提升了 29.36%和 43.93%。

3.4 小结

本章的主要工作是构建了一个基于噪声鲁棒语音识别实验平台，构建该平台的主要目的是用于噪声鲁棒性算法的验证与测试，以及后期指导嵌入式语音识别系统的实现。在构建过程中，首先是实验数据的准备，包括语音数据库和噪声数据库；其次是各个模块的设计与实现，包括系统参数配置、数据读入、前端处理、模型训练以及噪声鲁棒性测试；最后是基于该平台的语音增强鲁棒性实验。

第4章 快速特征变换算法和基于统计阈值的 CMVN

特征变换是特征级噪声鲁棒技术中的一种简单而又有效的噪声鲁棒技术，它使变换后的特征参数更加符合某种概率分布，并压缩了特征参数的值域范围、减小了训练和测试环境的不匹配。因此在语音识别系统，这些技术得到了广泛的应用。

在分块倒谱特征变换算法中，尽管其有效地解决了静音段过多、特征序列太长等问题，然而该算法对每一帧的局部区域都需要计算一次均值和方差，这也大大减慢了特征变换的速度。为了解决该问题，本文提出了一种递推的算法，能在线性的时间复杂度内快速地计算均值和方差。

另外本文在 CMVN 的基础上，提出了一种基于统计阈值的 CMVN，该算法不仅能滤除特征空间的高频噪声，还能进一步减小训练和测试环境的不匹配度。

4.1 分块倒谱特征变换递推算法

4.1.1 递推算法原理

由式(2.81)和式(2.82)可以计算出第 t 帧的均值和方差，也可以计算得出第 t+1 帧的均值和方差：

$$\boldsymbol{\mu}_{t+1}=\frac{1}{2N+1}\sum_{i=t-N+1}^{t+N+1}\mathbf{o}_i\ ,\quad \boldsymbol{\sigma}_{t+1}^2=\frac{1}{2N+1}\sum_{i=t-N+1}^{t+N+1}\left(\mathbf{o}_i-\boldsymbol{\mu}_{t+1}\right)^2 \tag{4.1}$$

从式(4.1)可以看出 $\boldsymbol{\mu}_t$ 与 $\boldsymbol{\mu}_{t+1}$，$\boldsymbol{\sigma}_t$ 与 $\boldsymbol{\sigma}_{t+1}$ 计算所用的数据只有两项不同，而且它们在物理位置相邻。

$$\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t=\frac{1}{2N+1}\sum_{i=t-N+1}^{t+N+1}\mathbf{o}_i-\frac{1}{2N+1}\sum_{i=t-N}^{t+N}\mathbf{o}_i=\frac{1}{2N+1}\left(\mathbf{o}_{t+N+1}-\mathbf{o}_{t-N}\right) \tag{4.2}$$

定义差分量

$$\Delta\boldsymbol{\mu}_{t+1}=\frac{1}{2N+1}\left(\mathbf{o}_{t+N+1}-\mathbf{o}_{t-N}\right) \tag{4.3}$$

代入式(4.2)得

$$\boldsymbol{\mu}_{t+1}=\boldsymbol{\mu}_t+\Delta\boldsymbol{\mu}_{t+1} \tag{4.4}$$

同理，方差也可以按照同样的方式递推得出：

$$\boldsymbol{\sigma}_{t+1}^2=\boldsymbol{\sigma}_t^2+\Delta\boldsymbol{\mu}_{t+1}\bullet\left(\mathbf{o}_{t+N+1}+\mathbf{o}_{t-N}-\boldsymbol{\mu}_{t+1}-\boldsymbol{\mu}_t\right) \tag{4.5}$$

根据以上推导，便可写出递推算法的程序代码，见图 4.1 所示。

```
// 输入参数：MFCCs 特征参数序列，frmNum 特征序列长度
// 输出参数：CMVNs 特征参数序列
void quickMfcc2Cmvn(FEATURE *CMVNs, FEATURE* MFCCs, int frmNum){
    for(int j=0; j<FEATURE_DIM; j++){   //最外层循环
        //step1: 递推参数初始化
        double mean=(N+1)*MFCCs[0][j];
        double std1=(N+1)*powl(MFCCs[0][j], 2);

        for(int t=1;t<=N;t++){
            mean+=MFCCs[t][j];
            std1+=powl(MFCCs[t][j], 2);
        }
        mean/=(double)(2*N+1);
        std1=std1/(double)(2*N+1)-mean*mean;
        CMVNs[0][j]=(MFCCs[0][j]-mean)/sqrtl(std1);

        //step2: 递推算法
        for (t=1;t<frmNum;t++){
            double deta, o1, o2;   //差分量，o(t-N), o(t+N+1)
            // 首尾边界处理
            if(t-N-1<0) { o1=MFCCs[t+N][j]; o2=MFCCs[0][j]; }
            else if(t+N>=frmNum) { o1=MFCCs[frmNum-1][j]; o2=MFCCs[t-N-1][j]; }
            else { o1=MFCCs[t+N][j]; o2=MFCCs[t-N-1][j]; }

            deta = (o1-o2)/(double)(2*N+1);     // 计算差分量
            mean+=deta;         // 递推计算均值
            std1+=deta*(o1+o2+deta-mean-mean);   //递推计算方差

            CMVNs[t][j]=(MFCCs[t][j]-mean)/sqrtl(std1);     //CMVN 特征输出
        }
    }
}
```

图 4.1 快速递推算法代码清单

4.1.2 递推算法分析和实验比较

表 4.1 中列出了两种方法每计算一次均值和方差所需的各种操作，可以看出当采用递推的方式时，加法、乘法以及访存次数比直接求取方法要大大减少，而且这些操作变得与区域大小 N 无关，更重要的是递推方法仅需要多使用一个额

外的存储，即存储差分量 $\Delta\boldsymbol{\mu}_{t+1}$。

表 4.1 快速 CMVN 算法和传统算法的性能比较

	加法	乘法	额外存储	访存次数	CMVN 平均性能 (微秒/帧)
式(2.81)、(2.82)	6N+3	2N+3	2	4N+2	467.26
式(4.4)、(4.5)	6	2	3	2	2.44

在表 4.1 中最后一列还列出了两种方法分别计算 CMVN 参数所用的平均时间，该实验结果是通过统计 10 000 条 39 维 MFCC 特征序列变换为 CMVN 特征而得（N=30），从实验结果上可以看出：采用递推方式求取 CMVN 的平均耗时远远小于传统方法。

4.2 基于统计阈值的 CMVN

当纯净语音信号受到噪声干扰时，其变化不仅体现在时域波形上，也体现在特征域上。在特征域上，表现为中心的偏移，特征值的范围变大，出现局部的异常点。对于中心偏移和特征值范围变大可以通过 CMVN 有效地解决，但对于局部异常点，CMVN 却显得无能为力。

4.2.1 统计阈值方法的基本原理

在离散信号处理中，若在局部区域出现异常点时，一般都将其视为高频噪声。为了能滤除这种噪声，一般都设定一个固定范围以限制信号的“溢出”，对于“溢出”部分用固定的边界值代替。但是对于语音信号的 MFCC 特征而言，其中心和值域范围总是随着不同的环境变化而变化，而设定一个固定阈值并不能适应环境的变化。因此，我们可以利用它的均值和方差给 MFCC 设定一个动态阈值：

$$\tilde{\mathbf{o}}_t[d]=\begin{cases}\mathbf{o}_t[d] & \left|\mathbf{o}_t[d]-\boldsymbol{\mu}_t[d]\right|\le\boldsymbol{\sigma}_t[d]\bullet T\\ \boldsymbol{\mu}_t[d]+\\ \quad \operatorname{sgn}\left(\mathbf{o}_t[d]-\boldsymbol{\mu}_t[d]\right)\bullet\boldsymbol{\sigma}_t[d]\cdot\bullet T & \text{其他}\end{cases} \tag{4.6}$$

式中 d 表示第 d 时间轨迹或第 d 维的 MFCC 特征，T 是一给定的阈值，sgn 是符号函数，见式(2.10)定义。式(4.6)说明动态范围大小由方差和阈值 T 决定，均值决定范围的位置。

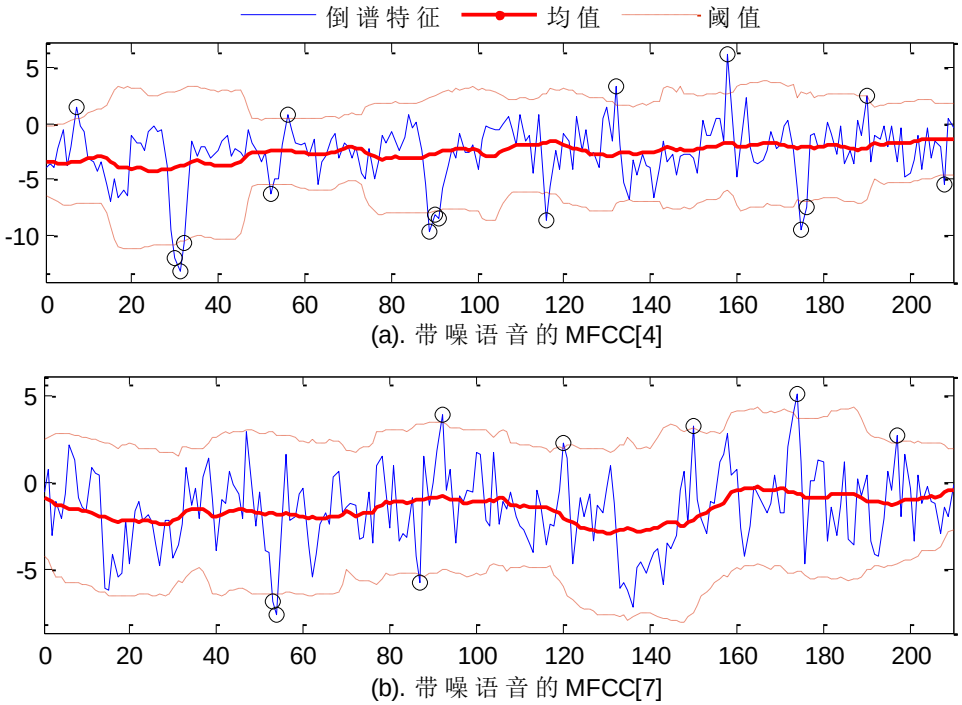

图 4.2 带噪语音的 MFCC 特征的动态范围

图 4.2 中显示了带噪语音（白噪声，*SNR*=0）的第 4 维和第 7 维 MFCC 特征和其动态阈值曲线，其中虚线表示动态阈值（*T*=2），当某一特征值偏离中心（均值曲线）太远并超出阈值范围时，便是认为是异常点（圆圈标识），此时需要用动态阈值代替这些异常点。

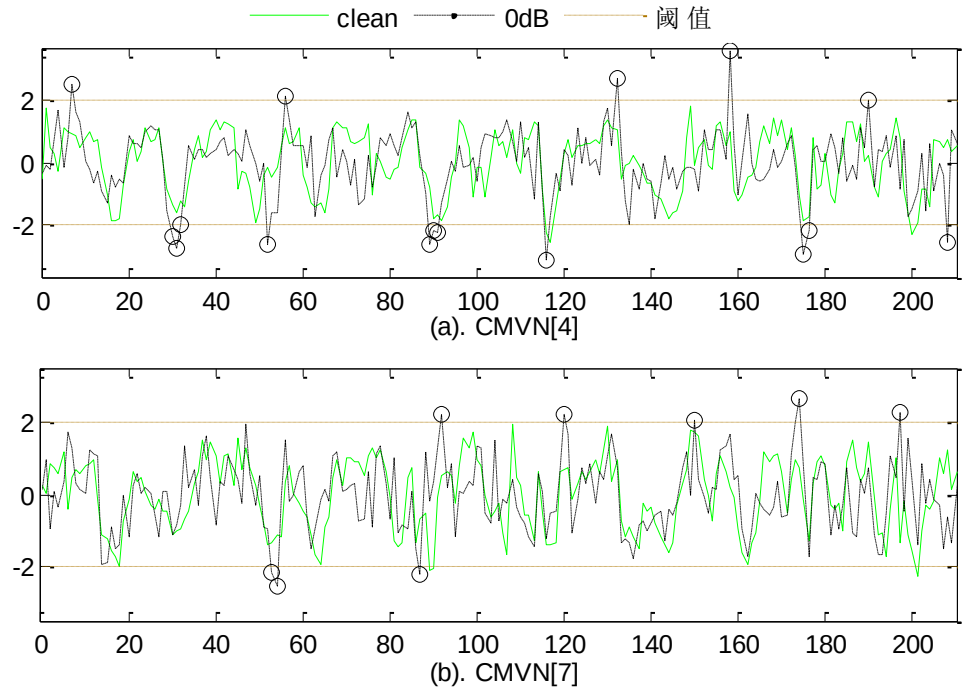

图 4.3 纯净语音的 CMVN 特征和带噪语音 CMVN 特征的阈值操作

若对式(4.6)进行 CMVN 处理后得

$$\breve{\mathbf{o}}_t[d]=\begin{cases}\dfrac{\mathbf{o}_t[d]-\boldsymbol{\mu}_t[d]}{\boldsymbol{\sigma}_t[d]} & if\ \left|\dfrac{\mathbf{o}_t[d]-\boldsymbol{\mu}_t[d]}{\boldsymbol{\sigma}_t[d]}\right|\le T \\ \mathrm{sgn}\left(\mathbf{o}_t[d]-\boldsymbol{\mu}_t[d]\right)\bullet T & \text{其他}\end{cases} \tag{4.7}$$

从上式可以看出：归一化处理后，原先的动态阈值现在变得位置固定（均值为 0），大小也固定（仅由 T 决定）。与式(2.84)相对比可知，式(4.7)仅比式(2.84)多一步阈值操作，因此我们称式(4.7)为基于统计阈值的 CMVN（Statistical Thresolding on CMVN，STCMVN）。

图 4.3 中是 CMVN 特征的阈值（T=2）操作，带噪语音的 CMVN 特征中的异常点用圆圈标出，它们将被阈值 T 或 $-T$ 代替，图中还给出了纯净语音的 CMVN 特征作对比，可以看出纯净语音的特征有时也会超出阈值范围。为了统一，因此在训练和识别时都需要经过阈值操作步骤。经阈值操作后，带噪语音的特征序列将更接近于纯净语音信号的特征序列。

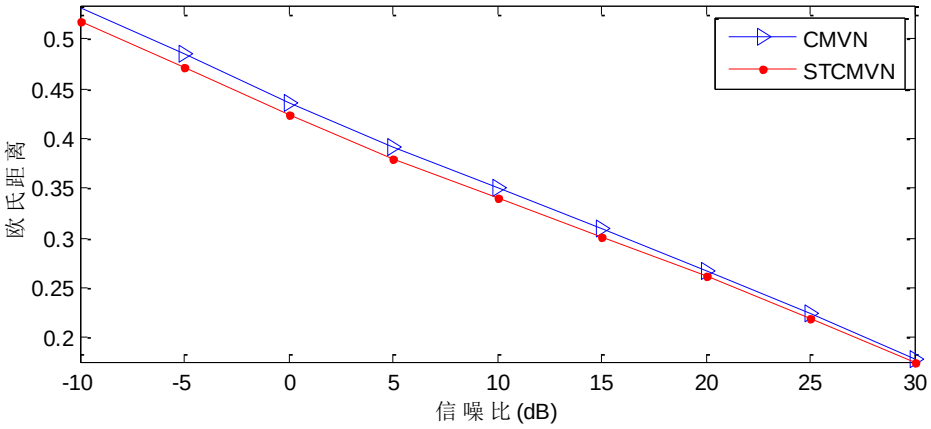

图 4.4 带噪语音和纯净语音特征参数平均每帧的欧氏距离

在图 4.4 中，三角形表示带噪语音和纯净语音 CMVN 参数之间的平均（每帧）欧式距离，实心圆点表示带噪语音和纯净语音 STCMVN 参数之间的平均（每帧）欧式距离，这些实验结果由 1 000 条语音数据分别在不同信噪比下得出(T=2)。从图 4.4 中可以看出：在不同信噪比情况下，STCMVN 参数的平均距离均小于 CMVN 的平均距离，也即使用 STCMVN 所产生的训练环境和测试环境不匹配程度要小于 CMVN。

4.2.2 阈值的确定

在使用 STCMVN 时，需要注意的一个问题是：当阈值 T 太大时，则不能起到滤除高频噪声的作用；当 T 太小时，会使许多有效的信息被滤除，降低了语音

信号特征参数的区分性。因此选择一个合理的阈值 T，成为 STCMVN 算法的关键。

在概率论中，切比雪夫不等式[63]（Chebyshev's inequality）描述了这样一个事实：在任何数据样本集中，绝大部分的样本都“接近”于其均值，或在任何概率分布中，绝大部分值都“接近”于该分布的均值。从数学角度可表述为：在任何分布中，样本值与均值的距离超过方差 σ 的 T 倍的概率不超过 $1/T^2$，即

$$P(|x-\mu| \geq \sigma T) \leq \frac{1}{T^2} \tag{4.8}$$

从上式可以看出，至少有 75.00%的数据与均值的距离小于 2 倍方差；至少有 88.89%的数据与均值的距离小于 3 倍方差；至少有 93.75%的数据与均值的距离小于 4 倍方差，因此可见绝大部分数据与均值的距离都很近。实验表明 T 的取值在 2~4 之间。

4.3 特征变换实验结果和分析

实验使用的语音库和噪声数据库与第三章语音增强实验中所使用的相同，实验的参数配置也相同。本次实验的目的是各种特征变换技术的实验结果对比，其中以 MFCC 为基线系统，分别与 CMS，CMVN 以及本文提出的 STCMVN（T=3.6）作对比。为了避免端点检测的影响，在本实验中测试语音的有效语音起止点使用其在无噪环境下检测到的起止点。

表 4.2 特征变换方法在无噪环境下的实验结果

	MFCC	CMS	CMVN	STCMVN
识别率（%）	97.24	98.48	98.29	98.38

表 4.2 中展示了测试语音在无噪情况下，各种不同特征变换方法的实验结果。可以看出 STCMVN 的识别结果略低于 CMS，并且高于 CMVN 和 MFCC，这是因为 STCMVN 和 CMVN 在进行特征变换时，会对特征参数造成一定程度的失真，从而使得识别率略有下降。

图 4.5 中展示了各种不同的特征变换方法在不同背景噪声、不同信噪比情况下的实验结果。可以看出对于较为平稳的 Volvo 噪声，这几种方法的识别结果在不同信噪比情况下相差不大，对于白噪声、粉噪声和人群噪声来说，在较高信噪比时（≥15dB）它们的识别结果也相差不大，STCMVN 的识别率有时略小于 CMVN；而在低信噪比时，CMVN 和 STCMVN 的优势更加明显，远远好于 MFCC 和 CMS，同时 STCMVN 要略高于 CMVN。实验结果表明在低信噪比情况下时，STCMVN 要比 CMVN 更具有鲁棒性。

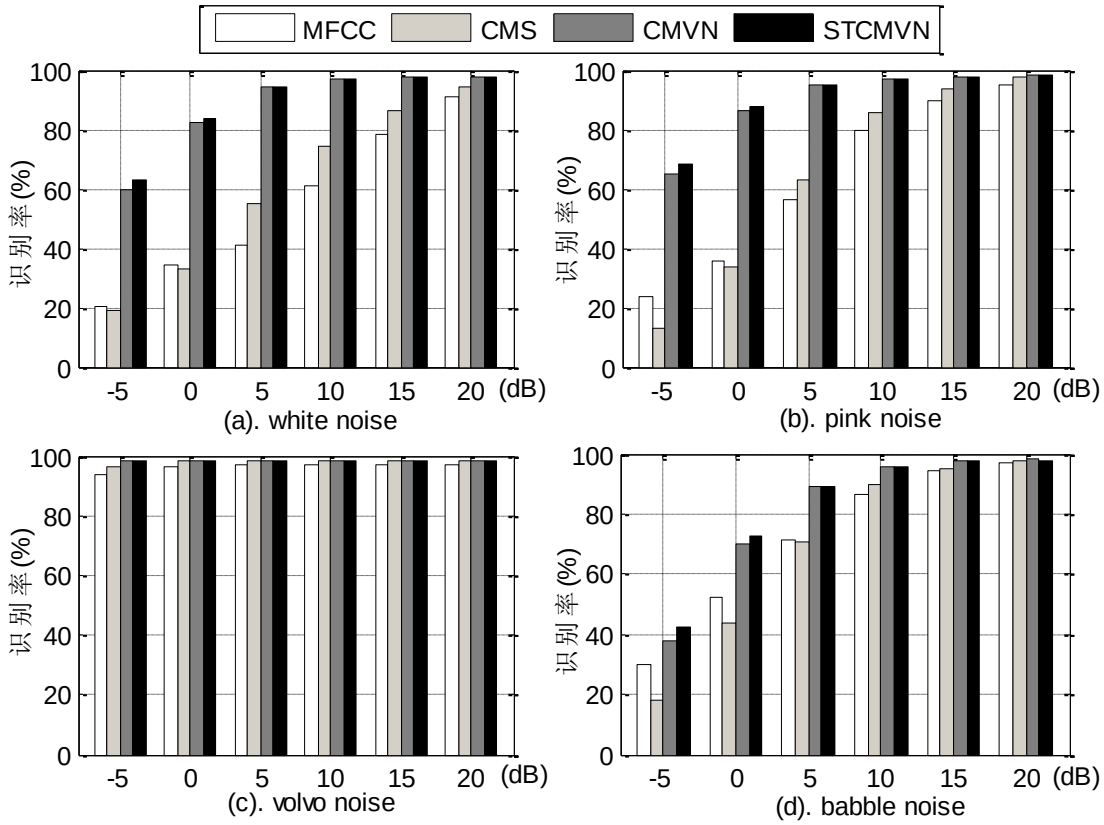

图 4.5 特征变换方法在不同噪声环境下的实验结果

4.4 小结

本章中为了降低 CMVN 参数的计算量，提出了分段 CMVN 的快速算法，通过递推方式来计算均值和方差。该快速递推算法不仅大大减少了乘法与加法运算次数，而且还大大减少了访存次数。实验结果表明，使用该快速递推算法，能使 CMVN 的计算速度得到显著的提升。

在本章中结合前面特征空间级噪声鲁棒算法中的特征变换，提出了基于统计阈值的倒谱均值归一算法。该算法继承了 CMVN 的优点，不但提供了环境独立的统计参数，而且还能简单有效地滤除特征空间的高频噪声，进一步减小了训练环境和测试环境的不匹配。文中还通过切比雪夫定理，确定了阈值 T 的取值范围。与 MFCC，CMS，CMVN 的特征变换实验表明：在低信噪比情况下，STCMVN 的噪声鲁棒性效果最好。

第5章 多种噪声鲁棒性算法的融合

前面主要讨论了两类噪声鲁棒性技术，即语音增强和特征参数变换。对于语音增强技术而言，它是从带噪语音信号中提取有用的语音信号，抑制、降低噪声干扰的技术；而特征参数级降噪主要通过对特征参数进行后处理，输出对噪声鲁棒的特征。在若将两种方法相结合结果会如何呢？

在本章中，将主要讨论语音增强和特征变换的融合算法。

5.1 语音增强与特征变换的两种融合算法

前面章节中的语音增强实验表明，语音增强算法能去除噪声，增加了语音的可懂度，但同时它也会残留音乐噪声对原始语音信号造成失真。因此，考虑到语音增强算法的利与弊，提出了两种融合算法。

方法 1：将增强后的语音仅用作端点检测（因为增强语音的端点检测结果比带噪语音的更准确），然后根据检测的结果取原始带噪语音中的“有效语音段”进行特征提取、特征变换等，这样避免了残留音乐噪声的影响。处理流程如图 5.1 所示。

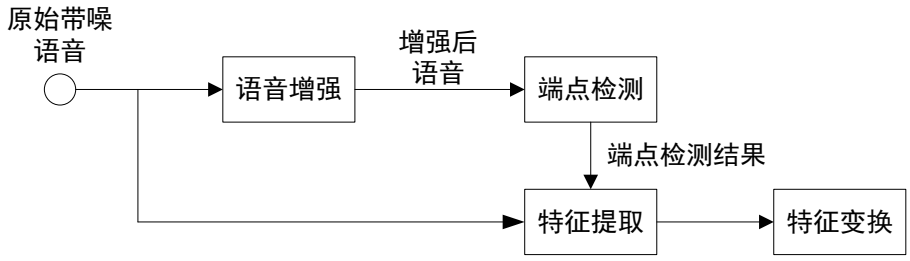

图 5.1 原始带噪语音用于特征提取

方法 2：将增强后的语音信号不仅用作端点检测，而且还用于特征提取、特征变换等，这样便能对语音增强的去噪特性加以利用，处理流程如图 5.2 所示。

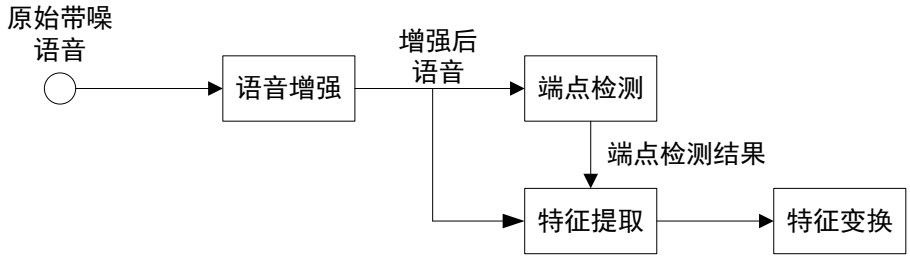

图 5.2 增强后的语音用于特征提取

5.2 实验结果和分析

实验所使用的语料库和噪声数据库与第 3 章语音增强实验所用的相同，参数配置与前面的语音增强和特征变换实验相同。

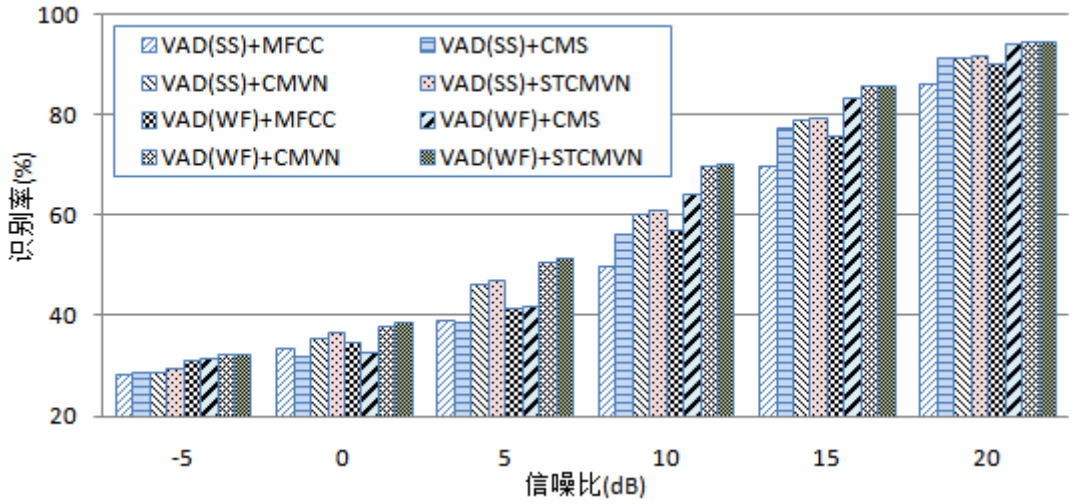

图 5.3 原始带噪语音用于特征提取的实验结果

图 5.3 中展示了语音增强算法用作 VAD 与各种特征变换方法相结合的实验结果。图中图例“VAD（SS）+MFCC”表示谱减法增强后的语音仅用于 VAD，特征提取采用原始带噪语音，特征参数是 MFCC，其他图例依次类推。

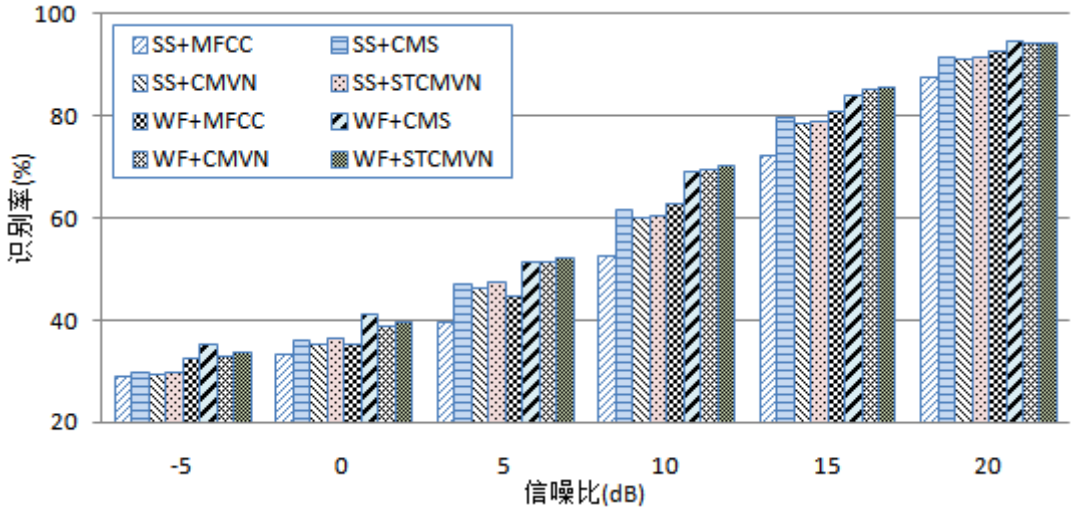

图 5.4 增强后语音用于特征提取的实验结果

图 5.4 中显示了增强后的语音既用于端点检测又用于特征提取的实验结果。语音增强算法采用 SS 和特征参数使用 MFCC 的图例表示为“SS+MFCC”。图 5.3

和图 5.4 的实验中使用了 2 种语音增强算法和 4 种特征参数，因此共 2×4=8 组实验结果。该实验数据是在不同噪声环境下（White，Pink，Volvo 和 Babble）得到的平均实验结果。

从整体看来，两组实验结果都随着信噪比不断地下降，识别率也不断地降低；CMVN 和 STCMVN（T=3.6）的实验结果要好于 MFCC 和 CMS 的实验结果；使用维纳滤波的实验结果要好于谱减法的实验结果。在较高信噪比时（≥15dB）CMVN 和 STCMVN 相差不大，有时 STCMVN 要略小于 CMVN，但在较低信噪比时（≤10dB）STCMVN 优于 CMVN。

在图 5.4 中可以看出，有时 CMS 的实验结果要好于 CMVN 和 STCMVN，这主要是因为语音增强算法的主要目的是去除加性噪声，这时使用 CMS 会对特征空间造成的失真较小。

表 5.1 各种噪声鲁棒性算法的平均识别率

SNR	-5	0	5	10	15	20
MFCC	22.55	33.00	40.65	53.02	72.50	87.60
CMVN	22.09	35.90	49.05	65.17	82.07	93.00
STCMVN	22.76	36.62	49.86	65.76	82.43	93.17
VAD(WF)+STCMVN	32.52	38.67	51.38	**70.24**	**85.84**	**94.60**
WF+STCMVN	**33.67**	**39.74**	**52.31**	70.17	85.74	94.34

表 5.1 中展示了 MFCC、CMVN、STCMVN 以及 STCMVN 与语音增强的两种融合算法的实验结果，表中的粗体数字表示在相同信噪比情况下各种方法中最好的实验结果。

从粗体数字的分布可以看出：最好的实验结果都出现在上述两种融合算法中；在较高信噪比时（SNR≥10dB），最好的实验结果分布于方法 1 中，而在较低信噪比时（SNR≤5dB），最好的实验结果分布于方法 2 中。这是主要是因为语音增强在去除噪声同时，也会抑制一些有用的语音信息和残留一些音乐噪声。这样在高信噪比时，带噪语音信号中所含的噪声较少，语音增强对原始信号造成的失真大于其去噪的能力，因此直接使用原始带噪语音进行特征提取和特征变换能取得较好的结果；在较低信噪比时，语音增强的去噪能力要大于其对信号造成的失真，因此使用增强后的语音做识别处理能取得较好的结果。

从表 5.1 中还可以看出，CMVN 的识别结果优于 MFCC，相对提升率在 SNR=10dB 时达到最高为 22.91%；STCMVN 的识别结果优于 CMVN，相对提升率最高高达 3.03%；STCMVN 与语音增强的两种融合算法总体上都优于仅使用 STCMVN，在低信噪比情况下（SNR≤5dB）时，相对识别率最高分别为 42.88% 和 47.90%，在较高信噪比情况下（SNR≥10dB）时，相对识别率最高分别为 6.80%

和 6.70%。

通过以上的实验分析可知，多种噪声鲁棒算法的融合能更好地提升语音识别系统在噪声环境下的鲁棒性。

5.3 小结

本章的主要内容是多种噪声鲁棒性算法的融合，其中主要讨论了语音增强和特征变换的两种融合算法。一种是语音增强后的信号仅用于端点检测，特征提取使用原始的带噪语音；另一种是增强后的语音信号用于整个识别过程。实验结果表明，这两种融合算法在整体上都要好于文中所提及到的其他未融合算法，这也说明多种噪声鲁棒算法的融合能更好地提升语音识别系统在噪声环境下的鲁棒性。

第6章 总结与展望

6.1 工作总结

本文主要针对语音识别系统中，由噪声造成的训练环境和应用环境的不匹配问题进行研究，也即是对语音识别系统噪声鲁棒性算法展开研究，以使汽车中的嵌入式语音识别系统能适应于汽车车内复杂的噪声环境。

论文采用了一个简化的 HMM 模型，即 NLP+GMM 来构建了一个语音识别系统。首先采用统计模型中的 GMM 对声学特征建模，适用于非特定人识别系统，有利于识别率的提高，并减少了模板库所占用的存储空间；其次采用 NLP 算法缩短了传统 HMM 解码的时间，有利于保证识别系统在嵌入式平台上的实时性。

在噪声鲁棒性方面，本文主要做了三个方面的研究。一是语音增强，介绍了两种常用的语音增强算法，即谱减法和维纳滤波，文中还讨论了这两种增强算法的一些改进算法，以减少音乐噪声对信号失真造成的影响。实验结果表明，在低信噪比（SNR=-5dB）情况下，谱减法和维纳滤波使得语音识别系统的识别率分别相对提升了 29.36%和 43.93%。

在噪声鲁棒性方面的第二方面研究是特征变换，文中介绍了常用特征变换方法 CMS 和 CMVN 的基本原理，并提出了基于统计阈值的 CMVN，简称 STCMVN。STCMVN 在 CMVN 的基础上滤除了一部分高频噪声，并进一步减小了训练环境和测试环境的不匹配。在特征变换实验中，CMVN 与 MFCC 相比，CMVN 的相对提升率最高达 24.03%；STCMVN 与 CMVN 相比，STCMVN 的相对提升率最高达 3.03%。

第三方面的噪声鲁棒性研究是多种噪声鲁棒性算法的融合，文中主要讨论了语音增强和特征变换的两种融合算法。第一种是增强后的语音只用于 VAD，而特征提取使用原始带噪的语音信号；第二种是增强后的语音即用于 VAD 又用于特征提取。对于 STCMVN 和维纳滤波相结合的实验表明，两种融合算法在整体上都要好于其他未融合的算法；在较低高信噪比情况（SNR≥10dB）下，两种融合方法相对于 STCMVN 分别最高提高了 6.80%和 6.70%，在较低信噪比情况（SNR≤5dB）下相对于 STCMVN 分别最高提高了 42.88%和 47.90%。对于这两种方法相比较而言，当信噪比较高时（SNR≥10dB）第一种方法要好于第二种；当信噪比较低时（SNR≤5dB）第二种方法要好于第一种。这主要因为在较高信噪比情况下，信号中含有的噪声较少，此时语音增强算法对原始信号造成的失真大于其去噪的能力，因此直接使用原始带噪语音进行特征提取能取得较好的识别

结果；而在较低信噪比情况下时，语音增强的去噪能力要大于其对信号造成的失真，因此使用增强后的语音进行特征提取能取得较好的识别结果。

本文中针对嵌入式的实时性要求，提出了快速 CMVN 算法，采用递推的方式求取 CMVN 特征。该算法不仅减少了加法和乘法的运算次数，还减少了访存次数，使得 CMVN 特征的计算速度显著提升。

6.2 未来展望

本文只是嵌入式语音识别系统中噪声鲁棒性的一个初探，在嵌入式语音识别系统中还有许多实际的问题需要考虑：

1. 本文采用的 NLP 算法对特征序列进行分段，该方法的分段的基本出发点是特征的变化量，相对于以音素为单位的分段方法而言，NLP 算法的物理意义变得不是很明确，因此可以在这一方面进一步突破。

2. 文中所提及到的语音增强算法都是在频域进行，时域信号变换到频域是一个非常耗费时间的过程，这样使用增强算法是会影响嵌入式系统的实时性。因此可以在时域语音增强算法做进一步研究或调整算法结构使得增强后频谱信息能直接应用到特征提取。

3. 文中采用 GMM 对声学特征建模，由于 GMM 的训练算法（EM 算法）复杂度较高，不利于嵌入式实现。因此可以在快速训练算法方面作进一步的研究，以满足在线训练或自适应。

4. VAD 的检测结果受的噪声影响很大，当 VAD 结果不准确时会造成分段的不准确，从而影响系统的识别率，因此需要对 VAD 算法作进一步的研究。

5. 在 STCMVN 算法中，尽管通过切比雪夫定理可以确定阈值的一个大致范围，但却没有一个确定最优阈值的方法。因此还需要进一步实验和研究。

致 谢

衷心感谢李银国教授和清华大学郑方教授两年来给我的悉心指导与关怀，这两位老师无论在学习、生活还是科研都给了我莫大的帮助，并且他们严谨求实的治学态度、精益求精的工作作风使我受益匪浅。在此，谨向两位老师致以最真诚的谢意和深深的敬意。

同时，感谢车载网络研究室的程安宇、徐洋等老师，感谢他们在实验室对我的悉心教诲和耐心指导。感谢汽车电子与嵌入式系统工程研究中心的各位老师，感谢他们在我研究生阶段的学习工作中给予我的指导和帮助。感谢我的辅导员夏淑芳老师，在我研究生学习生活中对我的关怀和帮助。感谢清华大学语音与语言研究中心的各位老师和同学，以及清华大学其他曾帮助我的朋友，是他们的一言一行让我深深体会到清华人“自强不息，厚德载物”的作风。

感谢语音组的郭皓婷、黄镭、薛雯、代生灿、杨雪梅、魏琴、欧阳西子、储雯、杨丽坤、胡方超等同学，是你们在我研究生阶段给予我支持和帮助，使我最终顺利完成我的课题任务。感谢实验室金辉、苗艳强、赵国庆、郭冬萧、廖钦渔、张玲、王晶莹等兄弟姐妹，因为有你们，我的研究生生活变得多姿多彩。感谢我的室友荣怡、舒适、任步庭，谢谢他们在学习和生活中对我的帮助和照顾。衷心感谢我的父母家人，是他们对我无私的关怀和帮助让我顺利完成学业。最后，再次向所有在工作学习和生活中给予过我帮助的老师、同学、亲人、朋友致以深深的谢意！

蒲甫安

2012 年 4 月 24 日

硕士期间从事的科研工作

主要从事的科研工作

“核高基”国家科技重大专项——汽车电子控制器嵌入式软件平台研发及产业化（2009ZX01038-002-002-2）

发表的论文

李银国, 蒲甫安, 郑方. Statistical Thresholding for Robust ASR [J]. 重庆邮电大学学报(自然科学版), 2012, 24(2): 127-132.

参考文献

[1] 蔡莲红, 黄德智, 蔡锐. 现代语音技术基础与应用[M]. 北京: 清华大学出版社, 2003.

[2] Xuedong Huang, Alex Acero, Hsiao-Wuen Hon. Spoken Language Processing: a guide to theory, algorithm, and system development [M]. New Jersey, USA: Prentice Hall, 2001.

[3] Rabiner & Juang. Fundamentals of Speech Recognition [M]. Prentice-Hall, 1983.

[4] 杨行峻, 迟惠生. 语音信号数字处理[M]. 北京: 电子工业出版社, 1998.

[5] 赵力. 语音信号处理[M]. 北京: 机械工业出版社, 2009.

[6] Atal A. Automatic recognition of speakers from their voices [J]. Proc. IEEE. 1976, 64:460-475.

[7] Reynolds, D A. An Overview of Automatic Speaker Recognition Technology Internat [C]. Conf. on Acoustics, Speech, and Signal Processing (ICASSP 2002), Orlando, Florida, 2002: 4072-4075.

[8] Marc A, Zissman. Comparison of Four Approaches to Automatic Language Identification of Telephone Speech [J]. IEEE Transactions on Speech and Audio Processing, 1996, 4(1): 31-44.

[9] 王炳锡, 屈丹, 彭煊. 实用语音识别基础[M]. 北京: 国防工业出版社, 2005.

[10] 蔡莲红, 贾珈, 郑方. 言语信息处理进展[J]. 中文信息学报, 2011, 25(6): 137-141.

[11] Wang D, Liu J, Liu R S, at al. Embedded speech recognition system on 8-bit MCU core [C]. Proceedings of the IEEE International Conference on Acoustics, Speech and signal Processing(ICASSP). 2004: 301-304.

[12] 陈振标. 嵌入式语音翻译系统的识别技术研究[D]. 北京: 中国科学院自动化所, 2004.

[13] 郭皓婷, 郑方, 罗灿华, 李银国. 嵌入式文本相关说话人识别算法的研究与开发[J]. 中文信息学报, 2010, 24(6):64-67.

[14] Canhua Luo, Xiaojun Wu, Thomas Fang Zheng, et al. Segmentation-based Method for Text-Dependent Speaker Recognition in Embedded Applications [C]. APSIPA , ASC, Singapore. 2010: 466-469.

[15] 杜俊. 自动语音识别中的噪声鲁棒性方法[D]. 中国科学技术大学, 2009.

[16] 雷建军, 杨震, 刘刚, 郭军. 噪声鲁棒语音识别研究综述[J]. 计算机应用研究, 2009(4): 1-6.

[17] 雷建军. 噪声鲁棒语音识别中的若干问题的研究[D]. 北京邮电大学. 2007.

[18] GONG Yifan. Speech Recognition in Noise Enviroments: A Survey [J]. Speech communication, 1955, 16(3): 261-291.

[19] ACERO A. Acoustical and Enviromental Robustness in Automatic Speech Recognition [D]. Pittsburgh: Carnegie Mellon University, 1900.

[20] 张仁志, 崔慧娟. 基于短时能量的语音端点检测算法研究[J]. 电声技术, 2005(7): 52-59.

[21] 刘华平, 李昕, 徐柏龄, 姜宁. 语音信号端点检测方法综述及展望[J]. 计算机应用研究, 2005, 25(8): 2278-2283.

[22] Khondaker A, Ghulam M. Improved Noise Reduction with Pitch-Enabled Voice Activity Detection [C]. ISIVC2008, Bilbao, Spain, July 2008.

[23] Boll, S F. Suppression of Acoustic Noise in Speech Using Spectral Subtraction [J]. IEEE Trans. on Acoustics, Speech and Signal Processing, 1979, 27: 113-120.

[24] Berouti M, R Schwartz, J Makhoul. Enhancement of Speech Corrupted by Acoustic Noise [C]. Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing, 1979: 208-211.

[25] Ch V Rama Rao, M B, et al. Speech Enhancement using a Modified Apriori SNR and Adaptive Spectral Gain Control [J]. International Journal of Computer Applications, 2011, 12(12).

[26] K K Paliwal, K K Wojcicki, B Schwerin. Single-channel speech enhancement using spectral subtraction in the short-time modulation domain [J], Speech Communication, 2010, 52(5): 450-475.

[27] 李银国, 薛雯, 徐洋. 基于噪声短时谱动态估计的语音增强谱减算法[J]. 重庆邮电大学学报（自然科学版）, 2010, 22(2): 127-130.

[28] V Tyagi, C Wellekens. On desensitizing the Mel-Cepstrum to spurious spectral components for Robust Speech Recognition [C], in Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP'05). IEEE International Conference on, 1: 529–532.

[29] Chen Yang, Frank K Soong, Tan Lee. Static and dynamic spectral features: Their noise robustness and optimal weights for ASR [C]. ICASSP, 2005: 241-244.

[30] Hermansky. Perceptual Linear Predictive (PLP) Analysis of Speech [J]. Journal

of the Acoustical Society of America, 1990, 87(4): 1738-1752.

[31] Alex A, Huang X D. Augmented cepstral normalization for robust speech recognition [C]. Proceedings of IEEE Automatic Speech Recognition Workshop, 1995: 146-147.

[32] Viiki O, Bye B, Laurila K. A recursive feature vector normalization approach for robust speech recognition in noise [C]. Proceedings of ICASSP'98, 1998.

[33] Huang J W, Shen J L, Lee S. New Approach for Domain Transformation and Parameter Combination for Improved Accuracy in Parallel Model Combination (PMC) Techniques [J]. IEEE Transactions on Speech and Audio Processing, 2001, 9(11): 842-855.

[34] Saga Yama S, Yamaguchi Y, Takahashi S, et al. Jacobian approach to fast acoustic model adaptation [C], In Proceedings of ICASSP, 1997: 2-4.

[35] 吕勇. 基于最大似然多项式回归的鲁棒语音识别[J]. 声学学报. 2010(1): 3-4.

[36] 郑方, 徐明星. 信号处理原理[M]. 北京: 清华大学出版社, 2003.

[37] 胡航. 语音信号处理[M]. 黑龙江: 哈尔滨工业大学出版社, 2002.

[38] Jyh-Shing Roger Jang. Audio Signal Processing and Recognition [EB/OL], available at the links for on-line courses at the author's homepage at http://www.cs.nthu.edu.tw/~jang.

[39] School of Physics Sydney, Australia, What is a decibel [EB/OL]. [2012-04-26]. http://www.animations.physics.unsw.edu.au/jw/dB.htm.

[40] Davis S B, Mermelstein P. Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences [J]. IEEE Transactions on Acoustic Speech and Signal Processing, 1980, 28: 357-366.

[41] 杨大利, 徐明星, 吴文虎. 语音识别特征参数选择方法研究[J]. 计算机研究与发展, 2003, 40(7): 963-969.

[42] 苏倩, 李银国. 基于 CHMM 语音识别特征参数的选择方法[J]. 计算技术与自动化, 2007, 26(4): 92-94.

[43] 边肇祺, 张学工. 模式识别（第二版）. 北京: 清华大学出版社, 1999.

[44] A P Dempster, N M Laird, D B Rubin. Maximum Likelihood from Incomplete Data via the EM algorithm [J]. Journal of the Royal Statistical Society, Series B, 39(1): 1-38.

[45] Max Welling, Learning Systems [EB/OL]. [2012-04-26]. http://www.vision.caltech.edu/welling/class/LearningSystems156B.html.

[46] J B Mac Queen. Some Methods for classification and Analysis of Multivariate

Observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability [M]. Berkeley, University of California Press, 1: 281-297.

[47] A Tutorial on Clustering Algorithms [EB/OL]. [2012-04-26]. http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/index.html.

[48] 郑方. 连续无限制语音流中关键词识别方法研究[D]. 北京: 清华大学计算机系, 1997.

[49] 徐明星, 郑方, 吴文虎. 基于 CDCPM 的语音识别拒识别方法[J]. 中国神经计算科学大会, 1997(2): 677-680.

[50] 郑方, 吴文虎, 方棣棠. CDCPM 及其在语音识别中的应用[J]. 软件学报, 1996, 863 高技术项目智能主题专刊(7): 69-75.

[51] Zheng F, Chai H X, et al. A real-world speech recognition system based on CDCPMs [C]. Int'l Conf. on Computer Processing of Oriental Languages (ICCPOL'97), Hong Kong, Apr 1997: 204-207.

[52] Juang B H, Rabiner L R. A probabilistic distance measure for hidden Markov models [J]. AT&T Technical Journal, 1985, 64(2): 391-408.

[53] 李鹏, 智强, 董明, 梁维谦, 刘润生. 嵌入式语音识别 Mahalanobis 距离计算模块[J]. 清华大学学报（自然科学版）, 2008, 48(07): 1202-1204.

[54] Noise Robust Speech Recognition, Microsoft Research [EB/OL]. [2012-05-19]. http://research.microsoft.com/en-us/projects/robust/.

[55] L D Alsteris, K K Paliwal. Short-time phase spectrum in speech processing: A review and some experimental results [J]. Digital Signal Processing, 2007, 17: 578-616.

[56] 胡广书. 数字信号处理—理论、算法与实现[M]. 北京: 清华大学出版社, 1996.

[57] 程佩青. 数字信号处理（第二版）[M]. 北京: 清华大学出版社, 2001.

[58] Chia-Ping Chen, Jeff A, Bilmes. MVA Processing of Speech Features [J]. IEEE Transactions on Audio, Speech, and Language Processing, 2007, 15(1): 257-269.

[59] Viiki O, Laurila K. Cepstral domain segmental feature vector normalization for noise robust speech recognition [J]. Speech Communication, 1998, 25: 133-147.

[60] Varga A P, Steeneken H J M, et al. The NOISEX-92 study on the effect of additive noise on automatic speech recognition [R]. Technical Report, Speech Research Unit, Defense Research Agency, Malvern, UK, 1992.

[61] NOISEX-92 噪声数据库[R], [2012-04-26]. http://spib.rice.edu/spib/select_noise.html.

[62] Audio Tool: Cool Edit Pro [R], [2012-05-19]. http://www.syntrillium.com.

[63] Chebyshev's inequality, Wikipedia [EB/OL]. [2012-03-11]. http://en.wikipedia.org/wiki/Chebyshev’s_inequality.