浙江科技学院学报,第 19 卷第 3 期,2007 年 9 月
Journal of Zhejiang University of Science and Technology
Vol. 19 No. 3, Sep. 2007

基于 Matlab 的语音端点检测实验研究

张震宇
(浙江科技学院 自动化与电气工程学院,杭州 310023)

摘　要:端点检测在语音识别中占有十分重要的地位,直接影响着系统的性能。今借助于 Matlab 这一功能强大的工具,成功地开展了语音端点检测的实验研究。首先简介端点检测涉及的几个基本概念,然后分析端点检测的基本方法,最后分别进行孤立字、孤立词的语音检测实验;重点阐述实验开展的具体过程,并给出部分关键源代码。实验取得了良好的效果。

关键词:端点检测;短时能量;过零率;Matlab
中图分类号: TP391.42　　**文献标识码**: A　　**文章编号**: 1671-8798(2007)03-0197-05

Experimental Study on Speech Endpoint Detection Based on Matlab

ZHANG Zhen-yu
(School of Automation and Electrical Engineering, Zhejiang University of Science and Technology, Hangzhou, 310023, China)

Abstract: Endpoint detection plays an important role in speech recognition, which directly affects performance of the speech system. With Matlab, experiments to detect speech endpoint are developed successfully. Firstly, several basic concepts are introduced briefly. Then, the basic method for endpoint detection is analyzed. At last, 2 experiments for isolated word are carried out. The detailed experiment procedure is focused on and part of key source codes is given, which gains favourable effect.

Key words: endpoint detection; short-term energy; zero-crossing rate(ZCR); Matlab

所谓端点检测,就是从一段给定的语音信号中找出语音的起始点和结束点。在语音识别系统中,正确、有效地进行端点检测不仅可以减少计算量和缩短处理时间,而且能排除无声段的噪声干扰、提高语音识别的正确率。研究表明,即使是在安静的环境下,语音识别系统一半以上的错误可能主要来自端点检测[1]。除此之外,在语音合成、编码等系统中,高效的端点检测也直接影响甚至决定着系统的主要性能。因此,端点检测的效率、质量在语音处理系统中显得至关重要,广泛开展端点检测实现手段方面的研究,有一定的现实意义。

笔者查阅了大量关于端点检测的文献资料,典型的如文献[2-5]等,发现大部分文献把重点放在理论分析层面上,集中研究了如何较好地改进检测方

收稿日期: 2007-04-23
作者简介: 张震宇(1976 —),男,浙江兰溪人,讲师,硕士,主要从事电子技术和语音信号处理的研究。

法;在利用 Matlab 工具实现时,大部分现有资料缺少详细的底层过程描述(如给出关键源代码),更多的是结果展示,这对于推广 Matlab 工具在语音信号处理领域的应用,显得较为欠缺。正是基于此,笔者利用 Matlab 进行了端点检测的基础实验,具体阐述了实验开展的过程,直接给出了部分关键的源代码,期望通过这些基础性的工作,能促进 Matlab 在包含端点检测在内的语音处理领域的普及应用。

1 几个基本概念[6]

1.1 短时性

从整体来看,语音信号的本质特征参数是随时间而变化的,所以它是一个非平稳态过程,不能用处理平稳信号的数字信号处理技术对其进行分析处理。从另一方面来看,虽然语音信号具有时变特性,但是在一个短时间范围内(一般认为在 10～30 ms 短时间内),其特性基本保持不变(即相对稳定),因而可将其看作是一个准稳态过程,即语音信号具有短时平稳性,这种特性称为语音信号的"短时性"。

1.2 短时能量(Short-Term Energy)

由于语音信号的短时性,因此对数字化后的语音信号一般需进行分帧处理,并认为 1 帧内信号的频谱特性和某些物理特征参量可近似看作不变。1 帧内的信号能量值称为"短时能量"。

设第 n 帧语音信号 $x_n(m)$ 的短时能量用 E_n 表示,则其计算公式为:

$$E_n = \sum_{m=0}^{N-1} x_n^2(m)$$

式中,N 为信号帧长。

1.3 过零率(Zero-Crossing Rate, ZCR)

1 帧语音信号中,信号波形穿过横轴(零电平)的次数,称为语音信号的"过零率"。定义语音信号 $x_n(m)$ 的过零率 Z_n 为:

$$Z_n = \frac{1}{2}\sum_{m=0}^{N-1} |\ \mathrm{sgn}[x_n(m)] - \mathrm{sgn}[x_n(m-1)]\ |$$

其中,sgn[ ] 是符号函数,即:

$$\mathrm{sgn}[x] = \begin{cases} 1, (x \geq 0) \\ -1, (x < 0) \end{cases}$$

为尽可能减少低频的干扰,在实际应用中往往对过零率做出简单的修正,即设定一个门限 T,将过零率的定义修改为穿越该门限的次数,即:

$$Z_n = \frac{1}{2}\sum_{m=0}^{N-1} \{ |\ \mathrm{sgn}[x_n(m) - T] - \mathrm{sgn}[x_n(m-1) - T]| + |\ \mathrm{sgn}[x_n(m) + T] - \mathrm{sgn}[x_n(m-1) + T]\ | \}$$

2 端点检测的基本方法

语音信号一般可分为无声段、清音段和浊音段。无声段是背景噪声段,平均能量最低;浊音段为声带振动发出对应的语音信号段,平均能量最高;清音段是空气在口腔中的摩擦、冲击或爆破而发出的语音信号段,平均能量居于前两者之间。清音段和无声段的波形特点有明显的不同,无声段信号变化较为缓慢,而清音段信号在幅度上变化剧烈,穿越零电平次数也多。经验表明,通常清音段过零率最大[7]。端点检测就是首先判断"有声"还是"无声",如果有声,则还要判断是"清音"还是"浊音"。为正确地实现端点检测,一般综合利用短时能量和过零率两个特征,采用"双门限检测法"。

现以孤立字"检"的发音为例,说明双门限检测法的原理,如图 1 所示。该方法需做出两级判断:首先利用浊音的短时能量最大的特点,由能量定位语音的大致位置。根据语音短时能量设定一个较高的门限 T_H,若信号的能量大于 T_H,则可确定 2 个端点 A、B,并可认为这 2 个端点之间是语音信号,这样相当于完成初判。再根据背景噪声的平均能量设定一个比 T_H 稍低的门限 T_L,如果信号的能量大于 T_L,则所对应的端点 C、D 之间仍是语音信号,至此完成了第一级判断。接下来进行第二级判断,由于语音的起点很可能是能量很弱的清音,此时还采用短时能量来区分清音和无声显然已不合适,应采用过零率。根据短时过零率设定一个新的较低门限 T,求越过该门限的过零率,从 C、D 两点分别向前、向后搜索,找到短时平均过零率与门限 T 的 2 个交点 E、F,这 2 个点就是语音的真正起点和终点[8]。

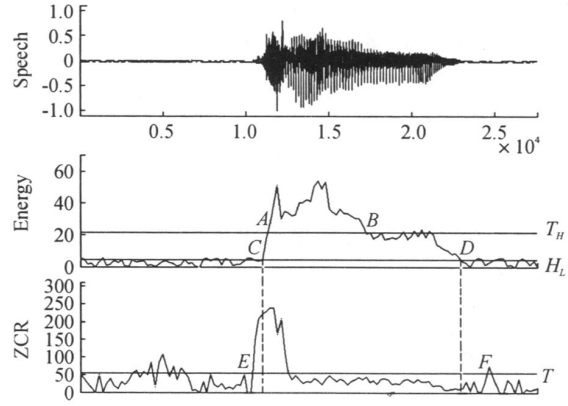

图 1 双门限检测法原理示意图

3　实　验

3.1　实验条件的准备

硬件上,PC 机配置为:815 主板、Celeron 1.0 GHz CPU、384 MB 内存、40 GB 硬盘、集成 AC97 声卡。软件上,为 Windows XP SP1 操作系统、Matlab 6.5 版本。

利用 Windows 系统自带的"录音机"程序,由麦克风输入笔者的原始语音,格式统一为"PCM 8 kHz, 8 位, 单声道",保存为.wav 文件。

3.2　程序设计

建立工作目录后,为完成实验任务,共编制 14 个程序(包括函数),其中最重要、最关键的是系统的界面设计和双门限算法设计。

3.2.1　界面设计　利用 Matlab 软件的图形界面设计工具 Guide,可以较容易地完成界面设计[9]。程序名为 epd_gui.m,菜单设计通过调用 3 个函数实现,即:

```
fig = createmadfig('epd');
fig = createmadloadmenu(fig);
fig = createmadmenus(fig);
```

按钮"显示曲线"、"显示门限"、"显示端点检测结果"等的部分设计代码在附录 1 中给出。

3.2.2　双门限算法设计　算法设计是端点检测的核心部分,根据上述的方法进行编程,程序名为 epdstats.m,其检测过程如图 2 所示。关键的部分代码在附录 2 中给出。

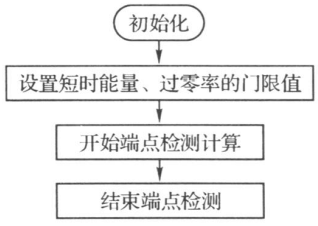

图 2　双门限检测法检测过程示意图

3.3　实验结果

笔者进行了大量孤立字和孤立词的语音实验工作,选取部分实验结果示于图 3、4。

可以看出,相比较于图 3,图 4 中语音信号波形有较明显的随机噪声,笔者认为,这可能是由于简陋的实验条件引起的:一是头戴式麦克风的质量较差,接收信号时不够稳定,和专业语音输入设备不能相提并论;二是室内无隔音等设备,实验室环境也不能做到完全的噪声限制。

上述实验结果较好地展示了"双门限检测法"的工作原理:首先利用短时平均能量门限值(先高后低)定位语音端点的大致位置,之后再利用短时过零率门限寻找端点的精确位置,从中可看出实验效果还是基本让人满意的。

限于实验条件,图 3 中信号的前段的短时过零率较大,使得程序在检测语音的前端时出现稍偏前的情况;图 4 中则出现信号后端稍偏后的情况。在一定程度上,可通过适当修改短时能量和短时过零率门限值来处理上述情况,但最本质的还是采用更优化的端点检测算法,以及改善实验条件等,这也是笔者在后续研究工作中值得继续深入的地方。

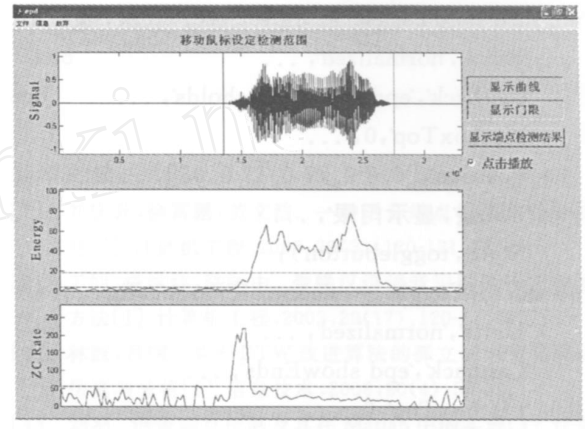

图 3　普通话发音,孤立字"九"的实验结果

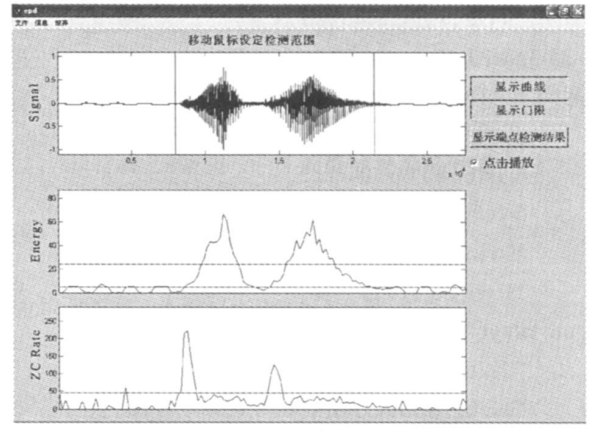

图 4　普通话发音,孤立词"浙江"的实验结果

4　结　语

本文详细阐述了基于 Matlab 工具进行孤立词语音端点检测的实验过程,并直接给出了部分关键

的源代码,实验中取得了较好的效果。因此,利用 Matlab 软件强大的运算能力和方便的编程风格,实现语音信号的端点检测是完全可行的。

附录 1:按钮设计代码

```
ud.plotsToggle = uicontrol('Parent',fig, ...
    'Units','normalized', ...
    'Callback','epd showPlots', ...
    'ListboxTop',0, ...
    'Position',[0.8 0.83 0.17 0.05], ...
    'String','显示曲线', ...
    'Style','togglebutton');
ud.thresholdToggle = uicontrol('Parent',fig, ...
    'Units','normalized', ...
    'Callback','epd showThresholds', ...
    'ListboxTop',0, ...
    'Position',[0.8 0.77 0.17 0.05], ...
    'String','显示门限', ...
    'Style','togglebutton');
ud.cursorToggle = uicontrol('Parent',fig, ...
    'Units','normalized', ...
    'Callback','epd showEnds', ...
    'ListboxTop',0, ...
    'Position',[0.8 0.7 0.17 0.05], ...
    'String','显示端点检测结果', ...
    'Style','togglebutton');
ud.Instructions = uicontrol('Parent',fig, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'ListboxTop',0, ...
    'Position',[0.08 0.95 0.66 0.04], ...
    'String','移动鼠标设定检测范围', ...
    'Style','text');
ud.playCheck = uicontrol('Parent',fig, ...
    'Units','normalized', ...
    'BackgroundColor',[0.8 0.8 0.8], ...
    'ListboxTop',0, ...
    'Position',[0.8 0.64 0.18 0.04], ...
    'String','点击播放', ...
    'Style','checkbox', ...
    'Value',1);
```

附录 2:算法设计代码

```
function [ITL,ITU,IZCT,energy,zc,N1,N2] =
epdStats(signal,fs)
winSizeMs = 10;          % 每帧设置为 10 ms
winShiftMs = 10;
winSize = millitosamples(winSizeMs,fs);
signedSignal = sgn(signal);
j = 1;
%计算短时能量和过零率
for i = 1:winShift:length(signal) - winSize
  energy(j) = (sum((abs(signal(i:i + winSize -
1)))));
  zc(j) = sum(abs(signedSignal(i + 1:i + winSize)
  - signedSignal(i:i + winSize - 1)));
  j = j + 1;
end
%设置短时能量门限值
IMX = max(energy);
IMN = mean(energy(silenceRange));
I1 = 0.03 * (IMX - IMN) + IMN;
I2 = 4 * IMN;
ITL = min(I1,I2);
ITU = 5 * ITL;
N2 = 0;
%function [N1,N2] = getEndPoints(ITL,ITU,
IZCT,energy,zc)
duration = length(energy);
backoffLength = length(1:winShift:millitosamples
(backoffMs,fs) - winSize);
done = 0;
%设置过零率门限值
IF = (((25 * winSizeMs)/10)/10000) * fs;
IZC = mean(zc(silenceRange));
zcstd = std(zc(silenceRange));
IZCT = min(IF,IZC + 2 * zcstd);
% 开始端点检测
for m = 1:duration
  if and(energy(m) >= ITL,~done)
    for i = m:duration
      if energy(i) < ITL
        break
      else
        if energy(i) >= ITU
```

```
            if ~done
               N1 = i - (i = = m);
               done = 1;
            end
            break
         end
      end
   end
end
startID = max(N1 - backoffLength,1);
endID = N1;
M1 = sum(zc(startID:endID) > = IZCT);
if M1 > = 3
   for i = startID:endID
      if zc(i) > = IZCT
         N1 = i;
         break;
      end
   end
end
done = 0;
```

% 结束端点检测

```
for m = duration: - 1:1
   if and(energy(m) > = ITL,~done)
      for i = m: - 1:1
         if energy(i) < ITL
            break;
         else
            if energy(i) > = ITU
               if ~done
                  N2 = i + (i = = m);
                        done = 1;
               end
               break
            end
         end
      end
   end
end
startID = max(1,N2);
endID = min(N2 + backoffLength,length(zc));
M2 = sum(zc(startID:endID) > = IZCT);
if M2 > = 3
   for i = max([1 startID]):endID % max added by
   MPC to prevent neg indices
      if zc(i) > = IZCT
         N2 = i;
         break;
      end
   end
end
```


参考文献:

[1] 刘庆升,徐霄鹏,黄文浩. 一种语音端点检测方法的探究[J]. 计算机工程,2003,29(3):120-121,138.

[2] 王朋,塔维娜,陈树中. 带噪汉语语音识别的端点检测方法[J]. 计算机工程,2003,29(17):120-121,135.

[3] 林波,吕明. 基于 DTW 改进算法的孤立词识别系统的仿真与分析[J]. 信息技术,2006,30(4):56-59.

[4] 刘羽. 语音端点检测及其在 Matlab 中的实现[J]. 计算机时代,2005(8):25-26.

[5] 郭继云,王守觉,刘学刚. 一种改进的基于频能比的端点检测算法[J]. 计算机工程与应用,2005,41(29):91-93,103.

[6] 赵力. 语音信号处理[M]. 北京:机械工业出版社,2005:31-37.

[7] 江官星,王建英. 一种改进的检测语音端点的方法[J]. 微计算机信息,2006,22(5-1):138-139.

[8] 张雄伟,陈亮,杨吉斌. 现代语音处理技术及应用[M]. 北京:机械工业出版社,2003:30-31.

[9] 张志涌. 精通 Matlab6[M]. 5 版. 北京:北京航空航天大学出版社,2003:238-242.