

语音端点检测及其在 Matlab 中的实现

刘 羽

(桂林工学院科技处, 广西 桂林 541004)

摘 要: 介绍了语音的基本特征和语音端点检测的基本方法, 对基于 Matlab 的端点检测算法进行了分析, 然后, 进行一个语音截取合并的实验, 结果表明: 用 Matlab 进行语音处理不仅编程简便, 也具有很好的效果。

关键词: 语音端点; 检测; 短时能量; 过零率; Matlab

1 引言

语音端点检测是语音分析、合成和识别中的一个重要环节, 目的是从包含语音的一段信号中找出语音的起始点及结束点, 从而只存储和处理有效语音信号。有效的端点检测不仅可以减少数据的存储量和处理时间, 而且能排除无声段的噪声干扰。端点检测的困难在于无声段或者语音段前后人为呼吸等产生的杂音、语音开始处的弱摩擦音或弱爆破音以及终点处的鼻音, 这些使得语音的端点比较模糊, 需要综合利用语音的各种信号特征, 从而确保定位的精确性, 避免包含噪音信号和丢失语音信号。本文讨论语音的特点、端点检测的一般方法及在 Matlab 中的实现方法, 最后给出一个简单的应用实例。

2 语音特点及端点检测方法

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。清音段为空气在口腔中的摩擦、冲击或爆破而发出的语音信号段, 平均能量居于两者之间。

采用基于能量的算法来检测浊音通常是可行也是可靠的。但对清音而言, 除非信号具有极高的信噪比, 例如在隔音室中录制的高保真度录音 (对于这种高信噪比录音, 最低电平语音的能量超过背景噪声能量几倍到几十倍), 否则, 采用能量算法从背景噪声中鉴别出清音就不够可靠了。

此时, 需要用到语音信号的另一重要特征, 即过零率: 一定时间内信号穿越零电平的次数。清音段与无声段的波形特点有明显不同, 无声段信号变化比较缓慢, 清音段信号由气流摩擦产生, 在幅度上的变化比较剧烈, 穿越零电平次数较多。经验表明, 通常清音段过零率最大, 无声段的过零率的变化范围较大。

可见, 振幅特征适合检测浊音, 过零率适合检测清音, 为了同时检测两者, 一般综合利用两种特征。

实际处理中, 由于语音信号是一种非平稳信号, 一般对数字化语音进行分帧处理, 并认为在一帧内它是平稳的。一帧内的信号能量值和过零次数被称为短时能量和过零率。下面分别描述它们的定义和检测方法。

2.1 短时能量

$$E = \sum_{n=1}^N |x(n)| \quad \text{或} \quad E = \sum_{n=1}^N x^2(n)$$

式中 $x(n)$ 为信号幅度, N 为语音帧长。

检测过程: 将短时能量与给定能量门限 $G1$ 相比, 若大于它并能在一定时间内达到门限 $G2$ 并维持给定帧数 (防止把短时脉冲误判为语音), 则认为是语音起点, 否则继续向下找。判定起点后, 继续将短时能量与给定门限 $G3$ 比较, 当小于它并能在一定时间内降到门限 $G4$ 并维持给定帧数 (防止将语音中的小停顿误判为语音结束), 认为是终点, 否则继续向下找。

2.2 过零率

$$ZCR = \sum_{n=1}^{N-1} |x(n) - x(n+1)|$$

求过零率时, 为避免无声段过零率太大, 一般还要设定一个门限 δ , 规定只有当相邻样点异号且差值大于 δ 时, 才将过零率数值加 1。

检测过程: 用短时能量方法找到语音起点后, 再计算起点之前几帧的过零率, 把求得的结果与给定门限比较, 连续 3 帧都大于门限则认为其中第一帧是语音起点。因为有些音节以清音开始, 单用能量方法检测可能会漏音, 而用过零率则可以避免这种情况。

端点检测可以采用上面描述的分别用短时能量和过零率与门限比较的双门限算法, 也可以把短时能量和过零率两者的乘积 (能频积) 作为检测参数, 此时只需进行单一门限比较。

3 基于 Matlab 的语音处理

Matlab (Matrix Laboratory) 是目前非常流行的科学计算和工程计算软件工具, 提供了一个高性能的数值计算、信号处理和可视化开发环境以及大量具有特殊用途的函数和工具箱, 其中包括基本的语音信号处理函数。

基本语音处理函数包括 wav 文件的读写函数, 声卡的录音和放音函数及波形的显示函数等, 还有一些第三方开发的语音处理工具 (如 Voice box 等)。利用这些函数和工具, 我们就可以方便地进行一些语音的处理工作。例如:

(1) 语音采样可以用以下命令实现:

$x = \text{wavrecord}(k * fs, 'dtype');$

其中 x 为语音采样信号, fs 为采样率, k 为采样秒数, 'dtype' 为采样数据类型。

(2) 语音数据也可以用以下命令从语音文件中读取:

$x = \text{wavread}('filename')$

(3) 语音信号的显示和回放可以用以下命令实现：

```
soundview(x,fs)
```

(4) 语音处理中，为了消除低频干扰通常要进行预加重滤波，数学表达式为 $1-aZ^{-1}$ ，其中 a 一般为 0.93~0.98，可用以下滤波命令实现：

```
x=filter([1 -a],1,x)
```

(5) 要计算短时能量，还要对信号进行分帧处理，可以使用 voice box 工具箱中的 enframe 函数：

```
y=enframe(x, FrameLen, FrameInc)
```

其中 x 为原始信号，FrameLen 为指定帧长，FrameInc 为指定帧移， y 为分帧后形成的二维数组，数组每行为一帧数据。

这里需要说明，voice box 为一个免费的第三方软件，需从网上下载后加入到 Matlab 的搜索路径才可使用。

(6) 短时能量可以简单地用以下命令实现：

```
amp1=sum(abs(x),2)
```

```
amp2=sum(x.*x,2)
```

(7) 过零率的计算相对复杂一些，可以用以下程序实现：

```
zcr = zeros(size(y,1),1);
```

```
delta = 0.02; % 设定的门限
```

```
for i=1:size(y,1)
```

```
    x = y(i,:);
```

```
    for j=1:length(x)-1
```

```
        if x(j)*x(j+1) < 0 & abs(x(j)-x(j+1))>delta
```

```
            zcr(i) = zcr(i)+1;
```

```
        end
```

```
    end
```

```
end
```

(8) 上面的程序用到了嵌套循环，运行速度较慢（Matlab 的弱点）。以下的矢量化算法更加适合 Matlab 的计算方式，具有较高的计算效率：

```
tmp1 = enframe(x(1:end-1), FrameLen, FrameInc);
```

```
tmp2 = enframe(x(2:end), FrameLen, FrameInc);
```

```
signs = (tmp1.*tmp2)<0;
```

```
diffs = (tmp1-tmp2)>0.02;
```

```
zcr = sum(signs.*diffs, 2);
```

该算法的巧妙之处在于 $tmp1$ 与 $tmp2$ 相差一个样点，因此当两者的乘积小于 0 时（第 3 行），表明相邻样点异号， $signs$ 取 1（即过零），否则取 0。而当两者的差值大于 0.02 时（第 4 行），表明相邻样点差值大于门限， $diffs$ 取 1，否则取 0。 $signs$ 与 $diffs$ 乘积之和即为同时满足两个条件的过零率结果。

不同场景的信号平均幅度变化很大，较难确定合理的最小门限，通常还需要先将输入信号进行归一化，将其幅度限制在 $[-1, 1]$ 之间，用以下命令实现 $x=x/\max(\text{abs}(x))$ 。

3 应用实例

本实例针对采访录音的整理进行设计。在采访中，由于被采访人对所提问题需要一定的时间进行思考，造成录音中无声（间隙）部分太多，内容不紧凑，也浪费空间。通过语音端点的检测，可以正确地识别出有效的语音部分，而将过长的无声部分和无声段中一些短暂的干扰信号剔除掉，使语音更为连贯。

设计程序界面采用 Matlab 的 GUI 技术。程序采用先录音后处理方式（非实时处理），录音长度预先设定，语音之间允许插入一定的静音间隔以保证可懂性，录音结果和处理结果都能够及时回放，并显示处理前后的波形。

采用双门限算法，对整个录音段连续检测，从而获得所有有效语音段，并加以剪切，合并，最后形成连贯的有效语音段。图 1 的例子中为 5.5 秒的录音，语音是“1-2-3-4”的发声，间隔约为 0.8 秒，取处理静音间隔为 0，处理结果波形如图 2 所示，时长为 1.48 秒，语音完整，没有断音现象。为了显示方便，这里只取了 5.5 秒的录音，实际应用中，时间可能会长得多，录音源也可能多种多样（如录音笔等数字设备），处理结果还可能需保存，因此涉及到文件的操作，此时，只需将录音函数改为读 WAV 文件函数（wavread）并增加写 WAV 文件函数（wavwrite）即可。

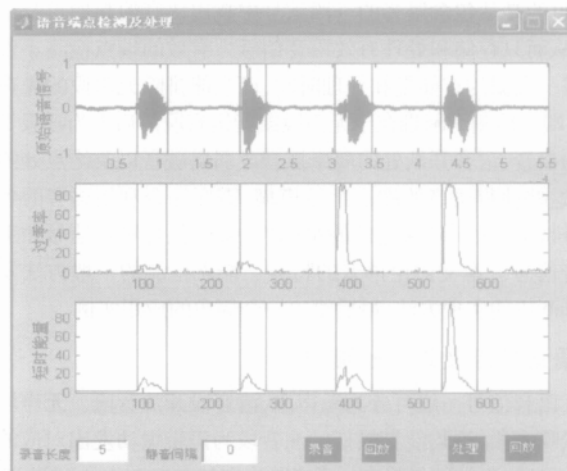


图 1 端点检测扫描结果

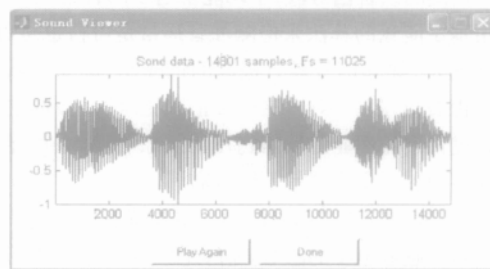


图 2 语音处理后回放结果

4 结束语

由于 Matlab 编程简便，计算中有很多现有的函数可以利用，更有一些免费的第三方工具的支持，结果可视化也简单可行，因此将其用于语音处理的各种算法实验，具有良好的效果。

参考文献：

- [1] [美]拉宾纳 L R, 谢弗 R W, 朱雪龙等译. 语音信号数字处理. 科学出版社, 1983.
- [2] 何强, 何英. Matlab 扩展编程. 清华大学出版社, 2002.
- [3] 果永振, 何遵文. 一种多特征语音端点检测算法及实现. 通讯技术, 2003. 1.
- [4] 李强, 赵伟. Matlab 数据处理与应用. 国防工业出版社, 2001.

