# 实验三 Voiced/Unvoiced/silence检测系统

## 1.实验目的

1.进行语音信号短时时域处理，了解短时能量、短时平均幅度和短时平均过零率。

2.画出清音/浊音的短时能量图，进行比较。

3.画出清音/浊音的平均过零率图，进行比较。

4.以一段语音为训练集，利用短时能量和短时过零率两种方法设定阈值，以另一段语音为测试集，得出测试语音的清浊音判断图（单位为帧）。

## 2.实验设备

应用软件Matlab2018a，四段录音，“Test.wav”,“清音f.wav”,“浊音b.wav”，“seaboat.wav”。

## 3.实验原理

### 3.1 短时能量和短时平均幅度

#### 3.1.1短时能量

设语音波形时域信号为x（n）,加窗函数w(n),分帧处理后得到的第i帧语音信号为yi(n),则yi(n）满足：

C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps1.jpg C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps2.jpg

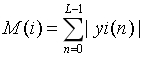
式中，w(n)为窗函数，一般为矩形窗或汉明窗；yi(n)是一帧的数值，n=1,2,......,L, i=1,2,....,fn,L为帧长；inc为帧移长度;fn为分帧后的总帧数。

计算第i帧语音信号yi(n)的短时能量公式为：

C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps3.jpg 1fn

#### 3.1.2 平均幅度

语音信号的平均幅度定义为:

 C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps5.jpg

M(i)也是一帧信号能量大小的表征，它与E(i)的区别在于计算时不论采样值的大小，不会因取二次方而造成较大的差异，在某些领域中会带来一些好处。

短时能量和短时平均幅度函数的主要用途有：区分浊音段和轻音段，因为浊音时E(i)值比轻音时大得多；区分声母和韵母的分界和无话段与有话段的分界。

### 3.2 短时平均过零率

短时平均过零率表示一帧语音中语音信号波形穿过横轴（零点）的次数。对于连续语音信号，过零即意味着时域波形通过时间轴；而对于离散信号，如果相邻的取样值改变符号，则称为过零。短时平均过零率就是样本数值改变符号的次数。

  短时平均过零率为

C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps6.jpg C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps7.jpg

式中，sgn[C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps8.jpg]是符号函数，即

C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps9.jpg

在实际计算短时平均过零率时，需要十分注意的一个问题是，如果输入信号中包含漂移，即信号在通往AD转换器前就有个直流分量，使AD转换后继续带有这个直流分量。因为直流分量的存在影响了短时平均过零率的正确估算，所以建议在语音信号处理前先消除直流分量。

理论上短时平均过零率是按上式计算，而在MATLAB编程中，却用另一种方法 。按上述过零的描述，即离散信号相邻的取样值改变符号，那它们的乘积一定为负数，即：

C:\Users\Suchu\AppData\Local\Temp\ksohtml3504\wps10.jpg

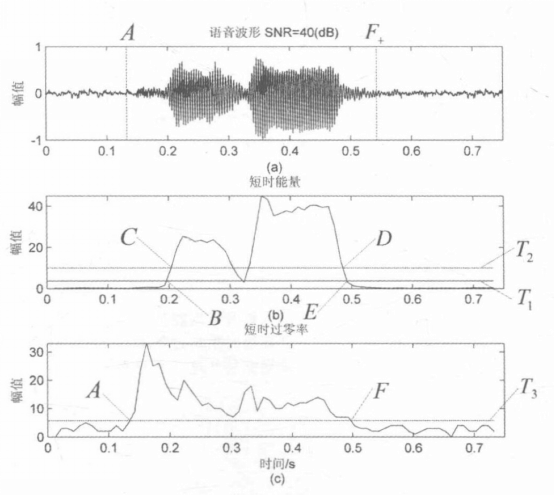
### 3.3语音端点检测原理

在语音信号处理中检测出语音的端点是重要的。语音端点的检测是指从包含语音的一段信号中确定出语音的起始点和结束点位置。因为在某些语音特性检测和处理中，只对有话段检测或处理。例如，在语音减噪和增强中，对有话段和无话段可能采取不同的处理方法；在语音识别和语音编码中同样有类似的处理。

处理没有噪声情况下的语音端点的检测，用短时平均能量就可以检测出语音的端点。但实际处理中语音往往处于复杂的噪声环境中，这时，判别语音段的起始点和终止点的问题主要归结为区别语音和噪声的问题。

双门限法

双门限法最初是基于短时平均能量和短时平均过零率而提出的，其原理是汉语的韵母中 有元音，能量较大，所以可以从短时平均能量中找到韵母，而声母是辅音，它们的频率较高，相应的短时平均过零率较大，所以用这两个特点找出声母和韵母，等于找出完整的汉语音节。双门限法是使用二级判决来实现的。



以上图为例，图（a）是语音的波形，图（b）是该语音的短时平均能量，图（c）是该语音的短时平均过零率。进行判决的具体步骤如下：

1.第一级判决

①根据在语音短时能量包络线上选取的一个较高阈值（门限）T2 （图中以虚水平线表示） 进行一次粗判，就是高于该T2阈值肯定是语音（即在CD段之间肯定是语音），而语音起止点应位于该阈值与短时能量包络交点所对应的时间点之外（即在CD段之外）。

②在平均能量上确定一个较低的阈值（门限）（图中以实水平线表示），并从C点往左、从D点往右搜索，分别找到短时能量包络与阈值T1相交的两个点B和E，于是BE段就是用双门限法根据短时能量所判定的语音段的起止点位置。

2.第二级判决

以短时平均过零率为准，从B点往左和从E点往右搜索，找到短时平均过零率低于某个阈值（门限）T3的两点A和F （图中T3以水平虚线表示），这便是语音段的起止点。

根据这两级判决，求出了语音的起始点位置A和结束点位置F。但考虑到语音发音时单 词之间的静音区会有一个最小长度表示发音间的停顿，就是在小于阈值T3满足这样一个最小长度后才判断为该语音段结束，实际上相当于延长了语音尾音的长度，如图中在语音波形图上标出语音的起止点分别为A和F+（从图中看出终止点位置为F，而实际处理中延长到 F+）。

在端点检测的具体运行中，首先是对语音分帧（实验二已做过介绍），在分帧基础上方能求出短时平均能量和短时平均过零率，然后逐帧地依阈值进行比较和判断。

### 3.4 Matlab语音处理基本指令语法说明：

#### 3.4.1 frame2time

我们在语音处理时常按帧来计算，例如计算短时能量、短时傅里叶变换，又有计算基音、计算共振峰都是以帧为单位，在作图时可以在横轴上标出是第几帧，但常常又希望横轴由时间来表示。一帧是一串样点，表示不同的时间，但用一帧时是表示一段时间内的特性，这时用一帧中的中点时间来表示为该帧的时间。frame2time函数就是计算每帧的中点的时间，方便作图时横轴用时间单位来表示。

功能；计算分帧后每一帧对应的时间。

调用格式：frameTime=frame2time(frameNum,framelen,inc,fs)

说明：输入参数frameNum是总帧数；framelen是帧长；inc是帧移；fs是采样频率。输出参数frameTime是每帧的时间，即取这一帧数据中间位置的时间。

创建frame2time的代码：

**function** frameTime=frame2time(frameNum,framelen,inc,fs)

% 分帧后计算每帧对应的时间

frameTime=(((1:frameNum)-1)\*inc+framelen/2)/fs;

#### 3.4.2 mean

M = mean(A)返回沿第一个数组维度的元素的平均值，A其大小不等于1。

M = mean(A,'all') 计算所有元素的均值A。

M = mean(A,dim)返回维度的平均值dim。例如，如果A是矩阵，则mean(A,2)是包含每行平均值的列向量。

如果A是向量，则mean(A)返回元素的均值。

如果A是矩阵，则mean(A)返回包含每列平均值的行向量。

#### 3.4.3 sum

S = sum(A)返回沿第一个数组维度的A元素的总和，其大小不等于1。

S = sum(A,'all') 计算所有元素的总和A。

S = sum(A,dim) 返回维度的总和dim。

如果A是向量，则返回元素的总和。sum(A)

如果A是矩阵，则返回包含每列总和的行向量。sum(A)

## 4.实验内容

1.画出清音和浊音的短时能量图，进行比较

2.画出清音和浊音的平均过零率，进行比较

3.以一段语音为训练集，利用短时能量和短时过零率两种方法 设定阈值，以另一段语音为测试集，得出测试语音的清浊音判断图（单位为帧）。

## 5.实验结果

### 5.1画出清音和浊音的短时能量图，进行比较

#### 5.1.1代码:

% 所用到的函数

% function frameTime=frame2time(frameNum,framelen,inc,fs)

% 分帧后计算每帧对应的时间

% frameTime=(((1:frameNum)-1)\*inc+framelen/2)/fs;

% end

clear all;

[x,fs]=audioread('D:\MYData\voice\sy3\清音f.m4a'); %读取音频

x=x(:,1); %提取一个通道

wlen=960; %设置窗长

win=hanning(wlen); %设置海宁窗

inc=400; %设置帧移

lx=length(x);

xf=enframe(x,win,inc)'; %分帧、加海宁窗

fnx=size(xf,2); %帧数

tx=(0:lx-1)/fs;

frametimex=frame2time(fnx,wlen,inc,fs);

**for** i=1:fnx

xf1=xf(:,i);

xe=xf1.\*xf1;

Enx(i)=sum(xe);

**end**

figure('Name','短时能量','NumberTitle','off');

subplot(2,2,1); %绘制清音信号时域图

plot(tx,x,'k');

axis([0 5 -0.6 0.6]);

xlabel('时间/s');

ylabel('幅值');

title('清音信号时域图');

grid;

subplot(2,2,3);

plot(frametimex,Enx,'k'); %绘制清音信号短时能量图

axis([0 5 0 15]);

title('清音短时能量');

xlabel('时间/s');

ylabel('幅值');

grid;

[y,fs]=audioread('D:\MYData\voice\sy3\浊音b.m4a');

y=y(:,1);

ly=length(y);

yf=enframe(y,win,inc)';

fny=size(yf,2);

ty=(0:ly-1)/fs;

frametimey=frame2time(fny,wlen,inc,fs);

**for** i=1:fny

yf1=yf(:,i);

ye=yf1.\*yf1;

Eny(i)=sum(ye);

**end**

subplot(2,2,2);

plot(ty,y,'k');

xlabel('时间/s');

ylabel('幅值');

title('浊音信号时域图');

grid;

subplot(2,2,4);

plot(frametimey,Eny,'k');

axis([0 5 -0.6 0.6]);

axis([0 5 0 15]);

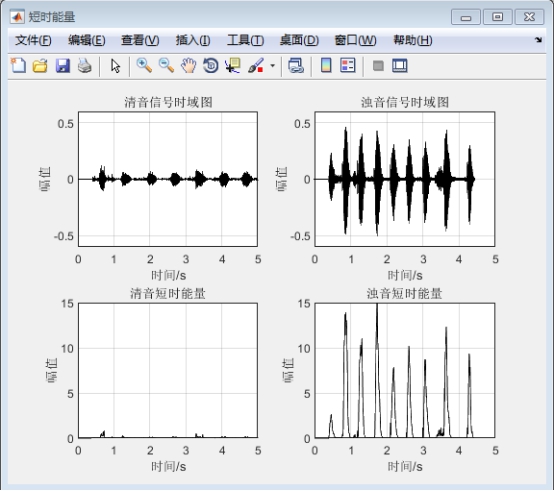
title('浊音短时能量');

xlabel('时间/s');

ylabel('幅值');

grid;

#### 5.1.2结果：



### 5.2画出清音和浊音的平均过零率，进行比较

#### 5.2.1代码：

% 所用到的函数

% function frameTime=frame2time(frameNum,framelen,inc,fs)

% % 分帧后计算每帧对应的时间

% frameTime=(((1:frameNum)-1)\*inc+framelen/2)/fs;

% end

clear all;

[x,fs]=audioread('D:\MYData\voice\sy3\清音f.m4a');

x=x(:,1);

wlen=960;

win=hanning(wlen);

inc=400;

xx=x-mean(x);

lxx=length(xx);

txx=(0:lxx-1)/fs;

xxf=enframe(xx,win,inc)';

fnxx=size(xxf,2);

zcrx=zeros(1,fnxx);

**for** i=1:fnxx

xxf1=xxf(:,i);

**for** j=1:(wlen-1)

**if** xxf1(j)\*xxf1(j+1)<0

zcrx(i)=zcrx(i)+1;

**end**

**end**

**end**

txx1=(0:lxx-1)/fs;

frametimexx=frame2time(fnxx,wlen,inc,fs);

figure('Name','短时平均过零率','NumberTitle','off');

subplot(2,2,1);

plot(txx,xx,'k');

axis([0 5 -0.6 0.6]);

xlabel('时间/s');

ylabel('幅值');

title('清音信号时域图');

grid;

subplot(2,2,3);

plot(frametimexx,zcrx,'k');

axis([0 7 0 350]);

title('清音信号短时平均过零率');

xlabel('时间/s');

ylabel('幅值');

grid;

%浊音

[y,fs]=audioread('D:\MYData\voice\sy3\浊音b.m4a');

y=y(:,1);

yy=y-mean(y);

lyy=length(yy);

tyy=(0:lyy-1)/fs;

yyf=enframe(yy,win,inc)';

fnyy=size(yyf,2);

zcry=zeros(1,fnyy);

**for** i=1:fnyy

yyf1=yyf(:,i);

**for** j=1:(wlen-1)

**if** yyf1(j)\*yyf1(j+1)<0

zcry(i)=zcry(i)+1;

**end**

**end**

**end**

tyy1=(0:lyy-1)/fs;

frametimeyy=frame2time(fnyy,wlen,inc,fs);

subplot(2,2,2);

plot(tyy,yy,'k');

axis([0 5 -0.6 0.6]);

xlabel('时间/s');

ylabel('幅值');

title('浊音信号时域图');

grid;

subplot(2,2,4);

plot(frametimeyy,zcry,'k');

axis([0 7 0 350]);

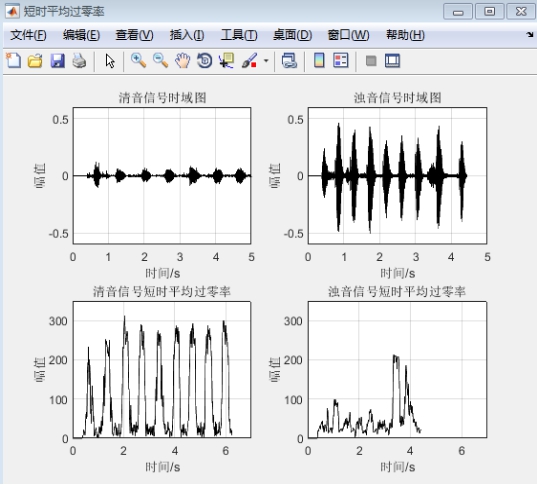
title('浊音信号短时平均过零率');

xlabel('时间/s');

ylabel('幅值');

grid;

#### 5.2.2结果：



### 5.3以一段语音为训练集，利用短时能量和短时过零率两种方法设定阈值，以另一段语音为测试集，得出测试语音的清浊音判断图（单位为帧）。

#### 5.3.1短时能量法

##### 代码:

clear all; clc;

[pyr,fs]=audioread('D:\MYData\voice\sy3\Test.wav'); % 读入数据文件 训练集

[pxr,fs]=audioread('D:\MYData\voice\sy3\seaboat.wav'); % 读入数据文件 测试集

pxr=pxr(:,1);

wlen=960; inc=400; % 给出帧长和帧移

win=hanning(wlen); % 给出海宁窗

pxr\_l=length(pxr); % 信号长度

pxrf=enframe(pxr,win,inc)'; % 分帧

pxr\_fn=size(pxrf,2); % 求出帧数

amp\_pxr=sum(pxrf.^2);

L=length(amp\_pxr);

pyr=pyr(:,1);

wlen=960; inc=400; % 给出帧长和帧移

win=hanning(wlen); % 给出海宁窗

pyr\_l=length(pyr); % 信号长度

pyrf=enframe(pyr,win,inc)'; % 分帧

pyr\_fn=size(pyrf,2); % 求出帧数

pyr\_t=(0:pyr\_l-1)/fs; % 计算出信号的时间刻度

**for** i=1:pyr\_fn

pyrf1=pyrf(:,i); % 取出一帧

pyrf1\_e=pyrf1.\*pyrf1; % 求出能量

En(i)=sum(pyrf1\_e);% 对一帧累加求和

**end**

pxr=pxr(:,1);

wlen=960; inc=400; % 给出帧长和帧移

win=hanning(wlen); % 给出海宁窗

pxr\_l=length(pxr); % 信号长度

pxrf=enframe(pxr,win,inc)'; % 分帧

pxr\_fn=size(pxrf,2); % 求出帧数

pxr\_t=(0:pxr\_l-1)/fs; % 计算出信号的时间刻度

**for** i=1:pxr\_fn

pxrf1=pxrf(:,i); % 取出一帧

pxrf1\_e=pxrf1.\*pxrf1; % 求出能量

Em(i)=sum(pxrf1\_e);% 对一帧累加求和

**end**

frameTime=frame2time(pxr\_fn,wlen,inc,fs); % 求出每帧对应的时间

subplot 211; plot(frameTime,Em,'k') % 画出短时能量图

title('短时能量');

ylabel('幅值'); xlabel(['时间/s' ]);

**for** i1 =1:L

**if** Em(i1)<=((max(En)-min(En))\*0.25);

c(i1)=0;

**elseif** Em(i1)<=((max(En)-min(En))\*0.5);

c(i1)=1;

**else**

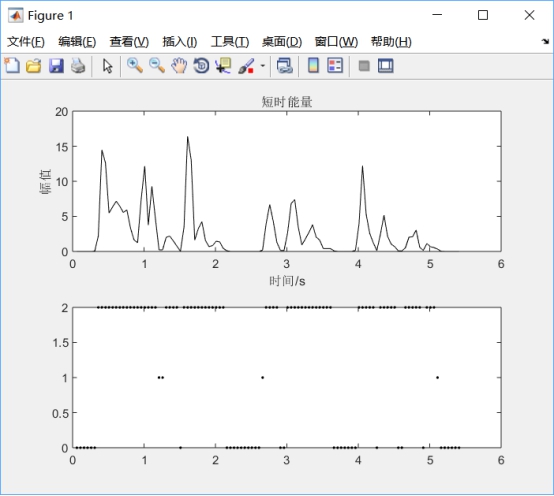
c(i1)=2;

**end**

**end**

subplot 212;plot(frameTime,c,'k.');

##### 结果：



(0: 0<=测试集能量<=0.25\*训练集能量

1：0.25\*训练集能量<测试集能量<=0.5\*训练集能量

2：0.5\*训练集能量<测试集能量)

#### 5.3.2短时过零率法：

##### 代码：

clear all; clc;

[x,fs]=audioread('D:\MYData\voice\sy3\Test.wav'); %练习集

[y,fs]=audioread('D:\MYData\voice\sy3\seaboat.wav'); %测试集

x=x(:,1);

wlen=960;

win=hanning(wlen);

inc=400;

xx=x-mean(x);

lxx=length(xx);

txx=(0:lxx-1)/fs;

xxf=enframe(xx,win,inc)';

fnxx=size(xxf,2);

zcrx=zeros(1,fnxx);

**for** i=1:fnxx

xxf1=xxf(:,i);

**for** j=1:(wlen-1)

**if** xxf1(j)\*xxf1(j+1)<0

zcrx(i)=zcrx(i)+1;

**end**

**end**

**end**

y=y(:,1);

yy=y-mean(y);

lyy=length(yy);

tyy=(0:lyy-1)/fs;

yyf=enframe(yy,win,inc)';

fnyy=size(yyf,2);

zcry=zeros(1,fnyy);

**for** i=1:fnyy

yyf1=yyf(:,i);

**for** j=1:(wlen-1)

**if** yyf1(j)\*yyf1(j+1)<0

zcry(i)=zcry(i)+1;

**end**

**end**

**end**

frameTime=frame2time(fnyy,wlen,inc,fs);

txx1=(0:lxx-1)/fs;

frametimexx=frame2time(fnxx,wlen,inc,fs);

subplot(2,1,1);

plot(frametimexx,zcrx,'k');

axis(0 6)

title('信号短时平均过零率');

xlabel('时间/s');

ylabel('幅值');

grid;

**for** i1 =1:fnyy

**if** zcry(i1)<=((max(zcrx)-min(zcrx))\*0.25);

c(i1)=0;

**elseif** zcry(i1)<=((max(zcrx)-min(zcrx))\*0.5);

c(i1)=1;

**else**

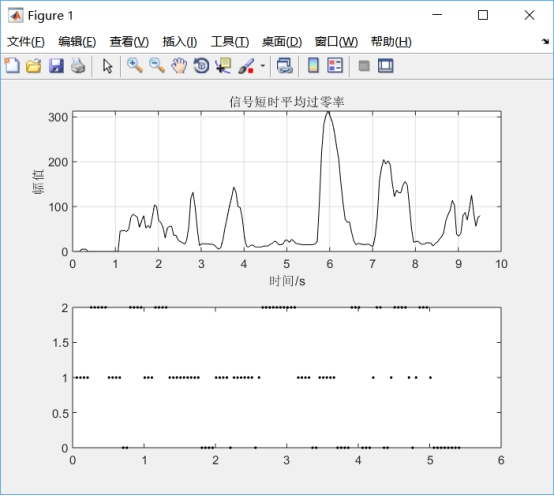
c(i1)=2;

**end**

**end**

subplot 212;plot(frameTime,c,'k.');

##### 结果：



(0: 0.5\*训练集短时过零率<测试集短时过零率

1：0.25\*训练集短时过零率<测试集短时过零率<=0.5\*训练集短时过零率

2：0<=测试集短时过零率<=0.25\*训练集短时过零率)