

## 一、基本介绍

Wavetave 的代码文档时通过 NaturalDocs 进行组织的

1、NaturalDocs 项目的主页：<http://www.naturaldocs.org/>

基本的用法可以参考 <http://www.naturaldocs.org/documenting.html> 举的例子。

2、Wavetave/doc/

下面有 NaturalDocs 生成的 HTML 文本介绍 WaveTave 工具箱的插件 Plugin 的用法

## 二、语音的特征提取

语音信号是非平稳的时变信号，是按分析帧(Frame)处理的，一帧的时间长度大约为 15ms~30ms。我们通过给语音信号加窗（矩形窗或 Hamming 汉明窗）获得分析帧。

（1）信号处理和语音信号的参考书籍

1、关于信号加窗处理的详细介绍，可以参考 A.V.奥本海姆的教材。

A.V.奥本海姆《离散时间信号处理》

A.V 奥本海姆《信号与系统》

2、关于语音信号处理的基本知识，可以参考赵力的教材。

赵力 《语音信号处理》

3、如果希望参考外国语音信号处理，可以参考 Thomas F.Quatieri 的教材

Thomas F.Quatieri 是正弦信号模型处理方法的提出者之一

《离散时间语音信号处理》 Thomas F.Quatieri

4、A.V.奥本海姆的信号处理的内容较为简单，如果需要涉及现代信号处理的书籍，可以参考张贤达的教材。

《现代信号处理》 张贤达 第二版 清华大学出版社

5、汉语语音信号处理

《汉语语音合成：原理和技术》 吕士楠 等著 2012 年 北京：科学出版社

（2）语音信号的基本描述参数

1、采样频率（Sample Rate）

8KHz：电话

16KHz：AM 广播

44.1KHz：CD 光盘

2、比特分辨率（bit resolution）

8bit

16bit

详细的标准内容参考 ITU 组织制定的标准 ITU-T G.728，ITU-T G.711

ITU 组织的网址: <http://www.itu.int>

### (3) 信号处理的需要特征

#### 1、信号的短时过零率、短时能量、短时自相关特征

当离散时间信号相邻两个采样点的符号相异时,我们称之为有一个过零点

#### 2、振幅(最大值、最小值、平均值),振幅的斜率(最大值、最小值、平均值)

#### 3、音频(Pitch)最大值、最小值、平均值,音频的斜率(最大值、最小值、平均值)

此处涉及到 Fundamental Frequency (F0)

#### 4、关于分析帧(Frame)也有相关的参数需要考虑

Frame Size: Frame 占据的点数

Frame hop size: Frame 与 Frame 之间的距离(起点距离起点)

Frame Overlap: 帧与帧之间可能会重叠

### (4) 同态信号处理需要以下的参数:

#### 1、梅尔倒谱系数(Mel-frequency cepstrum)

Mel-frequency cepstrum 的基本介绍可以参考语音信号处理的书籍。了解基本的同态信号处理和语音信号处理的基本知识即可。

### 三、src 文件夹下面的.m 文件

在 wavetave/doc 文件夹下面有通过 NaturalDocs 生成的,介绍了基本的情况,值得仔细看看。

以下是需要注意的东西

#### 1、VOT: Voice onset Time

可以参考维基百科的介绍

<http://zh.wikipedia.org/wiki/VOT>

[http://en.wikipedia.org/wiki/Voice\\_onset\\_time](http://en.wikipedia.org/wiki/Voice_onset_time)

#### 2、基频和音高的转换

[1]可以参考维基百科

<http://zh.wikipedia.org/wiki/%E9%9F%B3%E9%AB%98>

[2]中国知网 CNKI 上面也有一些文献介绍了此类的知识

《十二平均律音级(律音、音分)与频率的换算关系》高和 《乐器》 1984 年 04 期

### 四、OCT 文件夹下面的东西

《离散时间语音信号处理》 Thomas F.Quatieri

第九章 正弦模型

介绍了语音信号的正弦模型和随机确定模型的信息。

可以作为 OCT 文件夹下面的 cc 文件的参考。

### DeterministicSynth.cc

Synthesizes the deterministic component of the sound from Amp & Freq matrices.

### EnvelopeInterpolate.cc

Generates envelope from an array of heights and a given interval.

Returns an NDArray.

### FFTReflect.cc

Restore the symmetric structure of real fourier transform.

#### LinearPrediction.cc

线性预测算法

此处涉及到了 Levinson-Durbin 算法，与此算法等效的还有 Burg 算法等，具体内容可以参考《现代信号处理》 张贤达 第二版 清华大学出版社，张贤达的教材介绍了许多算法

代码中给出的连接

<http://www.emptyloop.com/technotes/A%20tutorial%20on%20linear%20prediction%20and%20Levinson-Durbin.pdf>

是英文的 Levinson-Durbin 算法，不过挺好读懂。

#### ParabolaInterpolate.cc

抛物线插值算法

定义了结构体

```
typedef struct
```

```
{
```

```
    double a0;
```

```
    double a1;
```

```
    double a2;
```

```
} Quadratic;
```

对于 STFT (Short Time Fourier Transform)，有两个文件进行了描述

STFTAnalysis.cc

STFTSynthesis.cc

如果手头上没有教材，可以参考网络上的开放课程或者课件，都是非常容易找到的。例如以下链接

<http://www.eee.hku.hk/~work3220/Speech%20analysis%20and%20synthesis-2%20STFT.pdf>

A.V. Oppenheim (奥本海姆) 的《离散时间信号处理》

John G. Proakis 的《数字信号处理》

均有介绍短时傅里叶变换 (Short Time Fourier Transform)

```
/* STFTAnalysis.cc
```

```
    Generate complex frames based on Short Time Fourier Transform.
```

```
*/
```

---

## 2.1 Short-Time Fourier Transform (STFT) Analysis

- Given time-series  $x[n]$ , the **STFT** at time  $n$  is given as:

$$X(n, \omega) = \sum_{m=-\infty}^{\infty} x[m]w[n-m]e^{-j\omega m}, \quad (2.1)$$

where  $w[n]$  is the **analysis window**, which is assumed to be non-zero only in the interval  $[0, N_w - 1]$ .

STFTSynthesis.cc

## 2.2 Short-Time Synthesis

- The **STFT** is highly redundant and it can be **inversed**.
- For each  $n$ , we take the inverse Fourier transform of  $X(n, \omega)$  from the **STFT**. We then obtain  $f_n[m] = x[m]w[n-m]$ . Evaluating  $f_n[m]$  at  $m = n$ , we obtain  $x[n]w[0]$ . Assuming  $w[0] \neq 0$ , we have  $x[n] = f_n[n] / w[0]$ . This gives to the following **synthesis equation**:

$$x[n] = \frac{1}{2\pi w[0]} \int_{-\pi}^{\pi} X(n, \omega) e^{j\omega n} d\omega. \quad (2.6)$$

- To reduce the computational complexity, the **STFT** is not computed at every time sample, but rather at a certain time-decimation rate. In some cases, the discrete **STFT** may not be invertible, i.e. there are certain constraints on the frequency-sampling and time-decimation rates.

```
/* SpectralEnvelope.cc  
  Extracts maximum heights from a magnitude spectrum.  
  Returns an NDAarray of spectral heights.  
*/
```

NDAarray 是 N-dimensional array 的简称

KlattFilter.cc 文件

在 20 世纪, 科技工作者发展了一系列的语音合成技术, 例如: Klatt Synthesizer, TD-PSOLA, FD-PSOLA, PSOLA-LPC

D.H. Klatt 的论文链接如下:

<http://sail.usc.edu/~lgoldste/Ling582old/Week%2012/klatt1980.pdf>

关于 Klatt Synthesizer 的教程可以参考:

<http://www.asel.udel.edu/speech/tutorials/synthesis/>

中文的书籍也有介绍此类原理：

《汉语语音合成：原理和技术》 吕士楠 等著 2012 年 北京：科学出版社

---

### 三、Until 文件夹

#### B2F.m

Bin-Index to frequency

大体的实现如下所示：

```
global FFTSize;
```

```
global SampleRate;
```

```
Ret = Freq / FFTSize * SampleRate;
```

#### DecibelToIFFTLn.m

将分贝（Decibel）转换成自然对数（Ln）的表现形式

此处涉及的是对数和分贝公式的变形

第一步，x 表示为指数和对数的组合

```
# x = e ^ ln(x)
```

第二部，对 x 求分贝的表示形式

```
# 20 * log10(x) = 20 * ln(x) / ln(10)
```

第三步，根据对数的换底公式，我们可以推导出

```
# 20 * log10(x) = 20 * ln(e ^ lnx) / ln(10)
```

第四步，这只是单位的换算，

分贝的单位为 dB， $20\ln x / \ln(10)$

```
# dB = lnx / ln(10) * 20
```

```
# lnx = dB / 20 * ln(10)
```

#### EpRIndexer.m

Linearly interpolates EpR parameter matrix for Variative EpR.

此处的 EpR 模型是第四部分的 MinCVE 的内容。

#### F2B.m

Frequency to Bin-Index

这和 B2F.m 是相对应的

#### GenerateSpectrum.m

```
# Calculates the decibel magnitude and phase spectrum from time domain
```

```
# signals.
```

这是生成频谱的问题

```
function [Ret, RetPhase] = GenerateSpectrum(Wave)
```

```
    global FFTSize;
```

```
    global Window;
```

```
    X = fft(fftshift(Wave .* Window));
```

这个 fft+fftshift 是经典的配合，可以参考 MATLAB 的 manual

```
Ret = abs(X)(1 : FFTSize / 2);
```

```
Ret = log10(Ret + 0.000001) * 20;
```

此处加入 0.000001 是确保精度，防止出现真数为 0 的情况（底数为 10，真数不为 0）

```
RetPhase = arg(X);
```

---

#### 四、MinCVE 文件夹

EpR Model 的全称为 Excitation Plus Resonance Model

此处部分，Sleepwalking 有开课：

【开坑】关于 EpR 语音模型的介绍、教程和讨论

<http://tieba.baidu.com/p/2874438644>

我个人不是很懂此处的内容，需要 SleepWalking 介绍