

AudioProcess

AudioProcess是一个能对音频进行简单处理的程序。

要运行此程序，可以打开MATLAB，运行根目录下的Mainapp.m，或是打开app文件夹下的Mainapp.mlapp再点击运行键。

根目录下AudioProcess.mlappinstall文件是打包后的app安装包，双击可安装到MATLAB app中。

由于音频录制的录制参数为采样率8000Hz、采样位数16、通道数1，可能会有语音失真出现。

如果不能正常运行，可能是没有将文件夹添加到MATLAB路径中，需要依次点击"主页"-"设置路径"-"添加并包含子文件夹"-选择AudioProcess文件夹-保存，然后再次尝试。

算法原理

语音生成模型

声管可以用若干级联的不等截面积均匀管道进行描述，每一节均匀管道的频响能够用一个单极点模型来近似，这样N段管道组成的声管就可以用一个N阶全极点滤波器表述，即

$$V(z) = \frac{G}{\prod_{k=1}^N (1 - p_k z^{-1})} = \frac{G}{1 - \sum_{k=1}^N a_k z^{-k}}$$

所有的极点 p_i 要分别构成共轭对以保证 $\{a_i\}$ 系数都是实数，再综合考虑清音信号，就可以得到产生语音信号的离散语音模型。

如果对上述模型进行简化，认为激励信号是一个脉冲序列，再去掉声门脉冲模型和口唇的辐射模型，就可以得到最简单的语音模型，如果假设激励信号 $e(n)$ ，语音信号 $s(n)$ ，根据全极点模型表达式，有

$$s(n) = \sum_{k=1}^N a_k s(n-k) + Ge(n)$$

语音预测模型

如果假设激励信号 $e(n)$ 是一个周期脉冲序列和一个高斯白噪声序列之和，就可以用一些信号处理方法求出 $\{a_i\}$ 系数，对语音生成模型进行变换，可得到

$$e(n) = s(n) - \sum_{k=1}^N a_k s(n-k)$$

语音重建模型

$$\hat{s}(n) = x(n) + \sum_{k=1}^N a_k \hat{s}(n-k)$$

如果激励信号 $x(n)$ 恰好等于 $e(n)$ ，那么重建语音 $\hat{s}(n)$ 和原始语音 $s(n)$ 完全相同。

语音的非平稳性导致预测系数 $\{a_i\}$ 系数是时变的，因此滤波过程要分段进行，每次用不一样的滤波器系数，最后再合成起来，并且要保持相邻两次滤波过程滤波器的状态不发生变化。

变速不变调

变速不变调是指声音播放时，速度的改变不会导致音调的变化。由语音预测模型，要实现变速不变调，就要从语音中分理出表示“调”的部分，即共振峰频率（预测模型的参数）和基音周期（激励信号的参数）。然后在不改变这两种参数的前提下增长或缩短数据的长度。最后用语音重建模型重建出变速后的语音。

由于预测系数 a_i 不适合直接用内插法得到缩放语音所要的信息，因此首先将预测系数变换为线谱对的归一化频率LSF，由于LSF反映了线性预测频域的共振峰特性，当语音从一帧往下一帧过渡时，共振峰会有所变化，LSF也会有所变化，且LSF参数可以直接内插。将LSF参数内插后，再逆变换回预测系数 a_i ，得到缩放后的预测系数。

具体的算法步骤：

1. 对原始输入语音进行预处理，进行分帧和归一化处理。
2. 计算每帧的预测系数 a_i 和信号增益 G
3. 对语音进行基音检测，得到基音频率和基音周期
4. 将基音周期按新语音的时长要求进行内插
5. 将预测系数 a_i 变换为对应的LSF参数，然后对LSF参数按时长要求进行内插，再逆变换回新的预测系数 a'_i ，此时新的预测系数满足新语音的时长要求
6. 根据缩放后的预测系数和基音周期用语音重建模型进行语音合成，得到变速后的语音

变调不变速

女声和男声的最大区别是频率高，一方面表现在基音频率高，另一方面表现在共振峰对应的谐振频率也更高。所以如果要升调，则应增加激励信号的频率，同时将共振峰的频率也响应增大一些，反之则降低。

第 i 个共振峰频率 F_i 可表示为

$$F_i = \frac{\theta_i f_s}{2\pi}$$

其中， θ_i 为全极点滤波器中第 i 个极点的幅角， f_s 为采样频率

从上式可看出，当基音频率改变时， θ_i 将改变，对应的共振峰频率也将发生变化。并且，对于不同的基音频率变化，不同的共振峰频率变化的数值是不同的，但为了简化起见，不论基音频率增加或减少多少，都将不同的共振峰频率增加或减少100Hz。

具体的算法步骤：

1. 对原始输入语音进行预处理，进行分帧和归一化处理。
2. 计算每帧的预测系数 a_i 和信号增益 G
3. 对语音进行基音检测，得到基音频率和基音周期
4. 根据升调还是降调确定是顺时针旋转根值还是逆时针旋转
5. 计算新的预测系数 a_i
6. 根据重新计算后的共振峰频率和基因频率用语音重建模型进行语音合成，得到变速后的语音

功率谱减法降噪

设语音信号的时间序列为 $s(n)$ ，加窗分帧（帧长 N ）处理后得到第 i 帧语音信号为 $s_i(m)$ ，对任意一帧信号做离散傅里叶变换后，有：

$$S_i(k) = \sum_{m=0}^{N-1} s_i(m) \exp\left(j \frac{2\pi mk}{N}\right) \quad k = 0, 1, \dots, N-1$$

如果已知噪声段的帧长为 NS ，则噪声段的平均能量为：

$$D(k) = \frac{1}{NS} \sum_{i=1}^{NS} |S_i(k)|^2$$

功率谱减法:

$$|\hat{S}_i(k)|^2 = \begin{cases} |S_i(k)|^2 - a \times D(k) & |S_i(k)|^2 \geq a \times D(k) \\ b \times D(k) & |S_i(k)|^2 < a \times D(k) \end{cases}$$

式中 a, b 为两个常数, a 称为过减因子, b 称为增益补偿因子, 改变 a, b 的值, 可改变降噪的效果

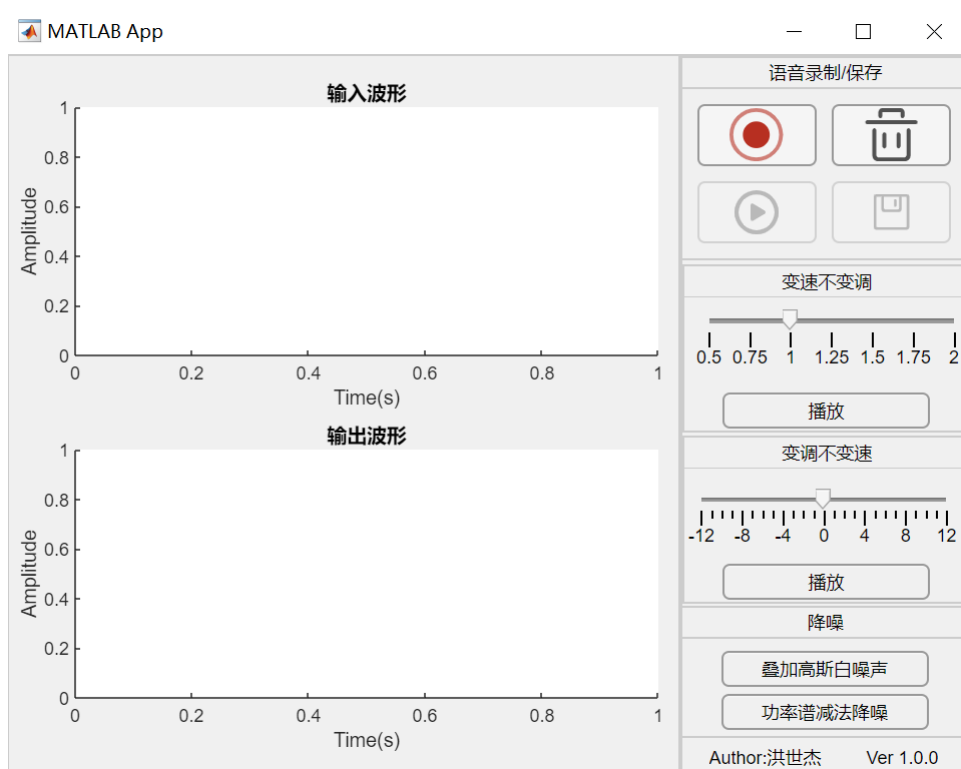
根据求得的 $|\hat{S}_i(k)|^2$ 和相位 $S_{\text{angle}}^i(k)$, 经过傅里叶反变换就可以求出降噪后的第 i 帧语音信号 $\hat{s}_i(n)$, 最后再经叠加合成就可以得到降噪后的语音信号 $\hat{s}(n)$ 。

具体的算法步骤:

1. 对原始输入语音进行预处理, 进行分帧和归一化处理。
2. 计算每帧离散傅里叶变换后的幅值和相位
3. 求出噪声段的平均能量
4. 对每帧进行功率谱减法
5. 傅里叶反变换得到降噪后的每帧语音
6. 将每帧语音叠加合成, 得到降噪后的语音信号

使用方法

程序界面



界面左栏为输入波形和输出波形, 界面右栏分为四个部分: 语音录制/保存、变速不变调、变调不变速和降噪。

语音录制/保存内共有四个按键, 从左到右, 从上到下分别为录制\停止录制、清空、播放和保存键。

录制\停止录制键: 点击录制键, 程序将调用系统麦克风进行音频录制, 录制参数为采样率8000Hz、采样位数16、通道数1。点击后按钮将变为停止录制键, 再次点击将停止录制。

清空键: 点击后将清空已录制的音频, 包括原始音频和处理后音频。

播放键：点击后将播放原始音频。

保存键：点击后将弹出保存选项框，通过选择要保存的音频类型和保存位置，将音频保存为wav格式



操作步骤

1. 点击录制键，记录语音音频；点击后图标将会变化，变为停止键，录制完成后，再点击一次，录制结束，输入波形和输出波形将会显示录制到的音频波形。
2. 点击播放键，播放记录到的原始音频。
3. 拖动变速不变调的滑动条，选择合适的倍速比例，松开左键后，将自动计算原始音频经变速不变调处理后的音频，并将波形显示在输出波形图中。点击播放键，可播放变速不变调处理后的音频。
4. 拖动变调不变速的滑动条，选择变调幅度，滑动条上的数字代表升高或降低的单音个数，根据十二平均律，一个完整八度中共有12个半音，因此如果拖动条指向12，则升高12个半音，即升高一个八度，对应音频的频率将变为原来的两倍，若拖动条指向-12，则降低12个半音，即降低一个八度，对应音频的频率将变为原来的二分之一。松开左键后，将自动计算原始音频经变调不变速处理后的音频，并将波形显示在输出波形图中。点击播放键，可播放变调不变速处理后的音频。
5. 点击“叠加高斯白噪声”按钮，将给原始音频叠加一个信噪比为5dB的高斯白噪声，并将处理后的波形显示在输出波形图中。然后，程序会自动播放叠加噪声后的音频。
6. 点击“功率谱减法降噪”按钮，程序将调用函数对叠加噪声后的音频进行降噪，并将处理后的波形显示在输出波形图中。然后，程序会自动播放叠加噪声后的音频。
7. 点击“保存”键，选择保存音频类型和保存位置，可对原始音频和处理后音频进行保存。

可能存在的Bug

1. 对于变速不变调和变调不变速功能，如果录制音频时间过短或者没有明显的语音，基音检测函数可能会返回空的基音频率和基音周期，导致后续语音合成过程报错。
2. 程序启动后，或是点击清空键后，尽管没有录制音频，你仍然可以拖动变速不变调和变调不变速功能的滑动条，但这会导致函数无法获取音频而报错。

主要功能API

变速不变调

实现文件：AdjustSpeed.m

function output = AdjustSpeed(ori_data,fs,speed_val)

Input:

ori_data:原始音频数据

fs:采样率

speed_val:速度倍率

Output:

output:处理后数据

可调参数

T1=0.1; r2=0.5; % 端点检测参数
miniL=10; % 有话段最短帧数
mnleng=5; % 元音主体最短帧数
wlen=240; % 窗长
inc=80; % 帧长

调整这些参数，对基音检测的结果有直接影响，进而导致最后的输出结果的好坏。不同的语音其最佳的参数各不相同。

变调不变速

实现文件: AdjustTune.m

function output = AdjustTune(ori_data,fs,Tune_val)

Input:

ori_data:原始音频数据

fs:采样率

Tune_val:音调升、降单音数，+12为升高十二个单音，即一个八度，对应频率 $\times 2$ （十二平均律）

Output:

output:处理后数据

可调参数

T1=0.1; r2=0.5; % 端点检测参数
miniL=10; % 有话段最短帧数
mnleng=5; % 元音主体最短帧数
wlen=240; % 窗长
inc=80; % 帧长

调整这些参数，对基音检测的结果有直接影响，进而导致最后的输出结果的好坏。不同的语音其最佳的参数各不相同。

叠加高斯白噪音

实现文件: AddNoise.m

function output = AddNoise(ori_data,snr)

Input:

ori_data:原始音频数据

snr:信噪比

Output:

output:叠加噪音后的音频数据

功率谱减法降噪

实现文件: PowerSpectrumSubtraction.m

function output = PowerSpectrumSubtraction(ori_data,fs)

Input:

ori_data:原始音频数据

fs:采样率

Output:

output:降噪后的音频数据

可调参数

a=4; % 过减因子

b=0.001; % 功率补偿因子

pre_time = 0.25; % 录音前段空白噪音的时长

调整这些参数，能改变降噪的效果。

工具函数

工具函数主要实现了基音检测，包括基音周期、基音频率等，关键函数是Ext_F0ztms.m。这些函数来源于《MATLAB在语音信号分析与合成中的应用》中的程序例程，部分函数根据参考资料内的原理分析和实际需要的功能进行了修改和优化。

函数名	功能
Ext_F0ztms.m	进行基音检测的关键函数
Ext_corrshtpm.m	延伸区间基音周期检测基本函数
Extoam.m	计算元音主体向过度区间延伸的长度
ACF_corrtpm.m	元音主体内用自相关函数法进行基音检测
back_Ext_shtpm1.m	前向延伸区间基音检测
fore_Ext_shtpm1.m	后向延伸区间基音检测
findmaxesm5.m	从自相关函数中提取5个峰值的幅值和位置（原函数为3个，根据参考材料改成5个）
findpeaks.m	寻找峰值幅值和位置
findSegment.m	按参数组合寻找每一组开始位置、结束位置和长度
pitch_vads.m	用能熵比法进行端点检测和元音主体检测
ar2lsf.m	将线性预测法计算的预测系数变换为LSF系数
lsf2ar.m	将LSF系数变换为线性预测法计算的预测系数
enframe.m	对数据进行分帧
ztcont11.m	对元音主体基音周期的前部进行修正
ztcont21.m	对元音主体基音周期的中部进行修正
ztcont31.m	对元音主体基音周期的后部进行修正
linsmoothm.m	信号线性平滑处理
pitfilterm1.m	结合线性平滑的中值滤波器
OverlapAdd2.m	在语音合成中用重叠相加法合成语音

参考资料

1. 程序界面的设计、交互控制、部分函数的编程和使用参考了[MATLAB帮助中心](#)
2. 功能的算法原理参考了课件PPT、参考资料和宋知用编著的《MATLAB在语音信号分析与合成中的应用》
3. 函数实现参考了参考资料和《MATLAB在语音信号分析与合成中的应用》中的程序例程
4. 工具函数参考了《MATLAB在语音信号分析与合成中的应用》中的程序例程
5. 程序内按钮图标来源于[iconfont-阿里巴巴矢量图标库](#)