

# 第二次大作业实验报告

zxdclyz

duskmoon314

BobAnkh

December 2020

## 1 原理与实现思路

### 1.1 思路简述

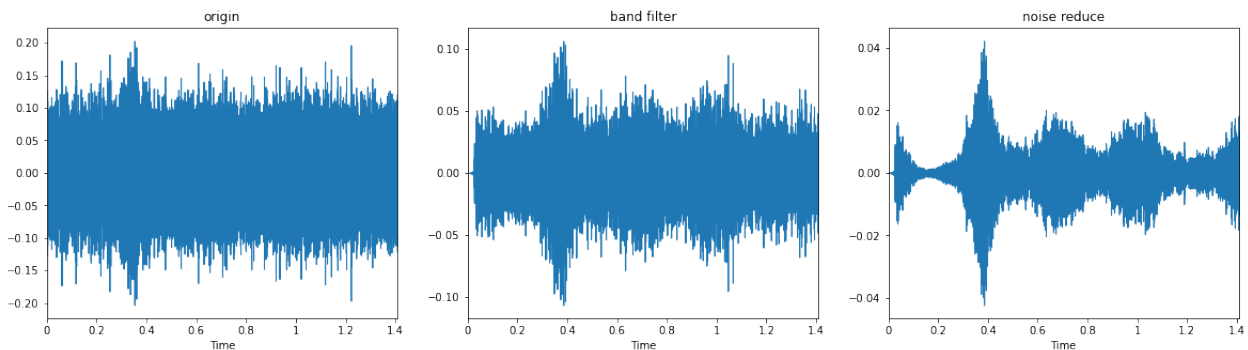
对于四通道音频信号，首先使用带通滤波和减谱法进行降噪，再以两个麦克风为一组计算六个互相关得到六个角度估计值，最后加权平均得到最终角度结果。

### 1.2 具体步骤

#### 1.2.1 噪声滤除

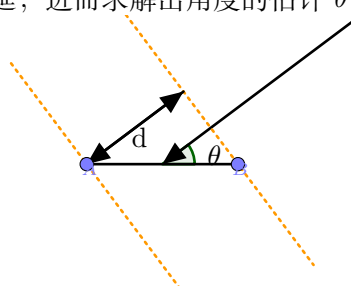
去噪的思路如下：

1. 查阅资料可知，女声的频率范围大致在 200Hz–1200Hz，在考虑泛音的基上，我们使用了 200–3000Hz 的带通 FIR 滤波器来滤除部分高频和低频噪声；
2. 通过带通滤波我们滤去了很大一部分噪声，但是通带内仍有一些噪声要处理。对于这部分噪声我们使用了谱减法处理，取音频信号的前一部分作噪声，再对音频分帧计算频谱，然后减去噪声谱，最后逆变换得去噪后的信号。



#### 1.2.2 分组计算角度

首先考虑两个麦克风接收平面波的情况。假设两个麦克风接收的平面波与麦克风连线夹角为  $\theta$ ，两个麦克风间距  $l$ ，则与时延对应的距离  $d = l \cos \theta$ ，时延  $\Delta t = d/v_{\text{声}}$ 。故可以对两个麦克风接收到的信号计算相关，取最大值和 0 的距离作为时延，进而求解出角度的估计  $\hat{\theta}$ 。



由于在声音角度与麦克风接近平行的时候  $\arccos$  的斜率比较大，会放大我们的计算误差，以至于在有些时候得到的  $\cos \theta$  值会超出  $-1 \sim 1$  的范围，尽管我们后面会舍去这样的角度，但为了防止特殊情况下由于这里的误差我们没有将它判定为误差较大的角，我们在此对边缘部分对  $\cos \theta$  进行了一个非线性的映射来限制它处于  $-1 \sim 1$  的范围并且适当减小了误差。

一共四个麦克风，两个一组可以得到六个角度估计值，即使简单的取平均也可得到比较不错的估计结果。下面介绍我们进一步优化使用的方法。

### 1.2.3 角度加权合并

如前所述，因为在计算  $\arccos$  时对于靠近-1 和 1 的三角函数值所计算出来的角度值误差比较大，所以在使用以上过程得到的 6 组角度数据时，为小于  $15^\circ$  和大于  $165^\circ$  的角度赋予权值 0（即不使用这样的角度来运算最终角度），为  $65^\circ - 115^\circ$  之间的角度赋予高权值，其余角度赋予低权值，从而加权平均后四舍五入得到最终角度。

### 1.2.4 采用多进程并行加速处理

因为考虑到测试集上需要处理的数据量比较大，为了加速运行，引入了多进程操作。使用标准库中的 `multiprocessing.Pool()` 类创建了进程池，来调用多个进程将处理任务映射到多核上并行处理，从而减少总体的处理时间。

## 2 结果展示与分析

使用训练集进行角度估计，结果展示如下：

语音序号	真实值	估计值	语音序号	真实值	估计值
1	315.000	315.000	8	84.000	84.000
2	358.000	358.000	9	119.000	119.000
3	107.000	107.000	10	349.000	349.000
4	154.000	154.000	11	218.000	218.000
5	48.000	47.000	12	107.000	107.000
6	201.000	201.000	13	297.000	297.000
7	303.000	303.000	14	109.000	109.000

可以看到，在训练集上我们的方法的准确性非常好，只有 5 号语音估计错误，且差错仅  $1^\circ$ 。

## 3 使用说明与文件清单

### 3.1 使用说明

1. 运行 `setup.py` 来安装需要的依赖包。
2. 以命令行方式运行 `main.py`，并使用 `-d` 参数来指定测试数据目录（默认值为 `test`），完整说明请参见 `-h` 参数。例如将测试数据集放在同目录 `test` 文件夹下，则可使用如下命令行来运行：

```
python main.py -d test
```

### 3.2 文件清单

表 1: 文件清单

文件名称	说明
<code>main.py</code>	主程序，按照上述使用说明在命令行中运行
<code>setup.py</code>	Python 3.6.12 下配置运行环境
<code>requirements.txt</code>	依赖库说明文件

## 4 总结与反思

我们在这次实验中最核心的方法仅仅是基于最简单的相关计算，也已经在这个问题上达到了不错的精度，可见相关这一计算工具的强大与实用性。本次实验中降噪部分也对我们的精度提示有不少帮助，这提示了我们对数据进行预处理的重要性。而有关不足，我们认为对于多个 mic（圆阵）的整体利用还可以加强，目前只利用了两两 mic 组合的加权求和。