信号与系统大作业《语音碎片我来拼拼听》
题目1_b 实验报告

1.实现思路

完成多段无污染语音拼接，可以考虑它们的相关函数，由于在matlab中，相关函数xcorr(a,b)相当于不同位移位卷积的结果，故而只要找到相关最大值就可以找到两端语音的对齐点。

2.实现方式

2.1第一版

代码已被替换，暂时无法提供。

思路是用y1，y2，y3三个向量来存储三段语音，分别做相关顺序实现拼接，写出文件。

此方法有明显缺陷，即代码无法复用，只能解决本小题的一个问题。

2.2第二版

与最终版代码基本相同，此处不做具体说明，与最终版的区别在于最终版将具体实现模块封装成函数，便于调用和复用。

2.3最终版

代码见codes中P1a的slice_stick.m文件。

思路是将所有的wav文件分为两个集合，正确集和错误集，首先先随机选取一个wav文件放入正确集(向量形式)，取正确集的首尾长度为'rate'的向量，与错误集中其余全部向量做最大相关，取这些最大值中的最大值，并比较首尾最大相关的大小，每次选取最"相关"的一个放入正确集中，并删去错误集的该文件，重复此操作，直到错误集长度为0。

此外，代码中有几处细节需要注意：

1.rate的长度与语音碎片受噪声污染程度有关，每小题此值略有差异。

2.代码中

```
r1 = mm_s/m_s;
r2 = mm_e/m_e;
m = min(abs(r1-1),abs(r2-1));%%%
```

此处m为最大相关归一化后与1的距离，不取max是因为对于一些特殊的语音，如前有空白的语音，r1或r2可能出现远大于1的情况。

3.本题中不存在交叠现象，但为了方便后续，设置了阈值，如果最大相关与1的距离超过0.2则不在添加语音。这个阈值也与语音片段有关。

3.源码截图

主程序：ex1.m

```
clear all; clc; close all;

file_path='P1a/';
out = slice_stick(file_path,24000);
audiowrite('P1a.wav',out,8000);
```

函数：

```
function all_audio = slice_stick (dic,len)
%初始化所有wav文件，随机找出一个放入正确集
file_list = dir(strcat(dic,'*.wav'));
l = length(file_list);
out_name = file_list(1).name;
out_file = audioread(strcat(dic,out_name));
delete (strcat(dic,out_name));
file_list = dir(strcat(dic,'*.wav'));
l = length(file_list);
rate = len; %length of the start/end
%选取相关最大值，放入正确集
while l>0
    len=length(out_file);
    start = out_file(1:rate);
    edge = out_file(len-rate+1:len);
    r_s = xcorr(start);
    m_s = max(r_s);
    r_e = xcorr(edge);
    m_e = max(r_e);
    tag = -1;
    max_f = 10;
    max_pos = 0;
    max_name = '';
    for x = 1:l
        f_name = file_list(x).name;
        file = audioread(strcat(dic,f_name));
        rr_s = xcorr(start,file);
        rr_e = xcorr(edge,file);
        mm_s = max(rr_s);
        mm_e = max(rr_e);
        %归一化
        r1 = mm_s/m_s;
        r2 = mm_e/m_e;
        m = min(abs(r1-1),abs(r2-1));%%%
        if m<max_f
            max_r1 = r1;
            max_r2 = r2;
            max_name=f_name;
            max_f=m;
            if(abs(r1-1)<abs(r2-1))
                tag = 1;
                max_pos = find(rr_s==mm_s);
            else
                tag = 0;
                max_pos = find(rr_e==mm_e);
            end
        end
    end
    if(max_f<0.2)%%%阈值限制
        tmp=audioread(strcat(dic,max_name));
        tmp_l=length(tmp);
        if(tag==1)
            out_file = [tmp(1:tmp_l-max_pos);out_file];
        else
            out_file = [out_file;tmp(tmp_l-max_pos+rate+1:tmp_l)];
        end
    end

    delete (strcat(dic,max_name));
    file_list = dir(strcat(dic,'*.wav'));
    l = length(file_list);
end
all_audio = out_file;
```

4.实验结果及分析

P1a.wav文件，观察波形和人耳试听后，发现结果应该正确。

本题是解决其余问题的基础。