

哈爾濱工業大學

实验报告

题目	语音信号端点检测
----	----------

专业	计算机科学与技术
----	----------

学号	1160300329
----	------------

班级	1603107
----	---------

学生	黄海
----	----

指导教师	郑铁然
------	-----

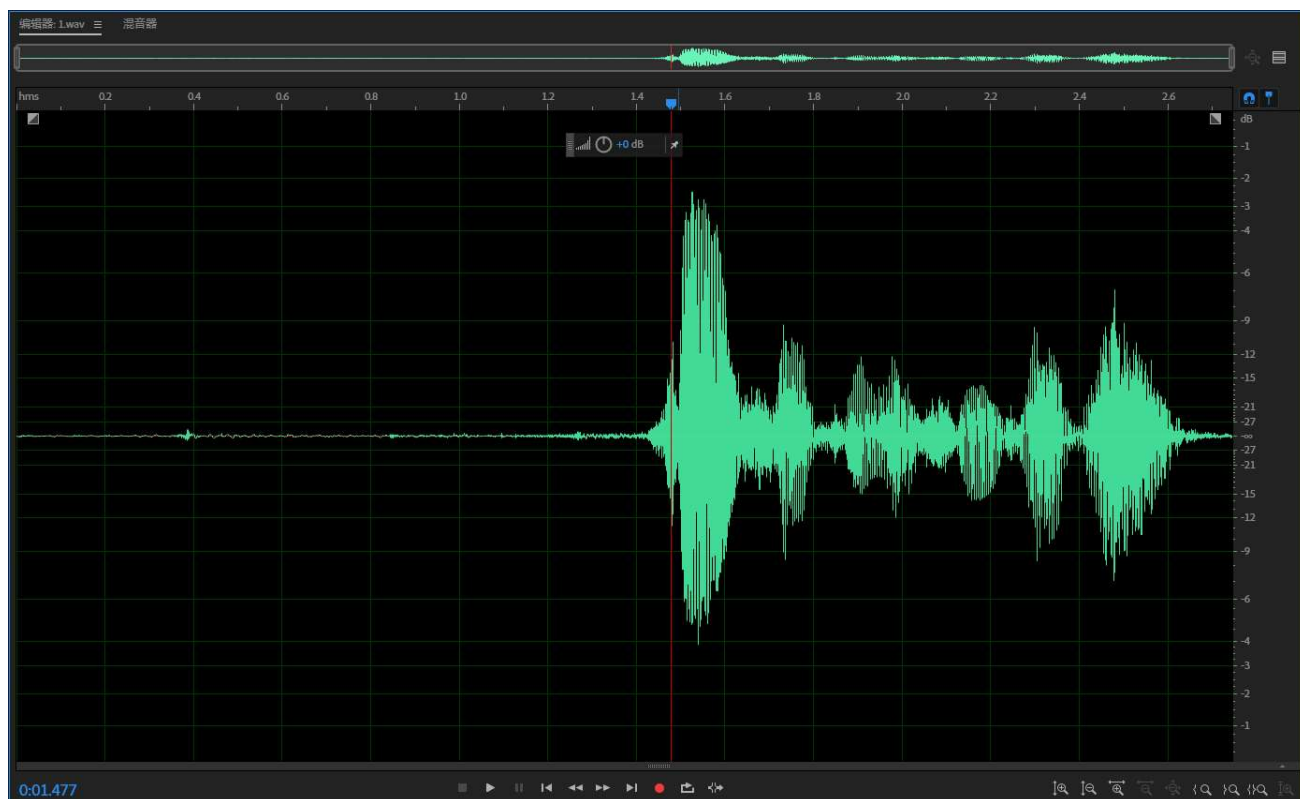
实验地点	G709
------	------

实验日期	10.15
------	-------

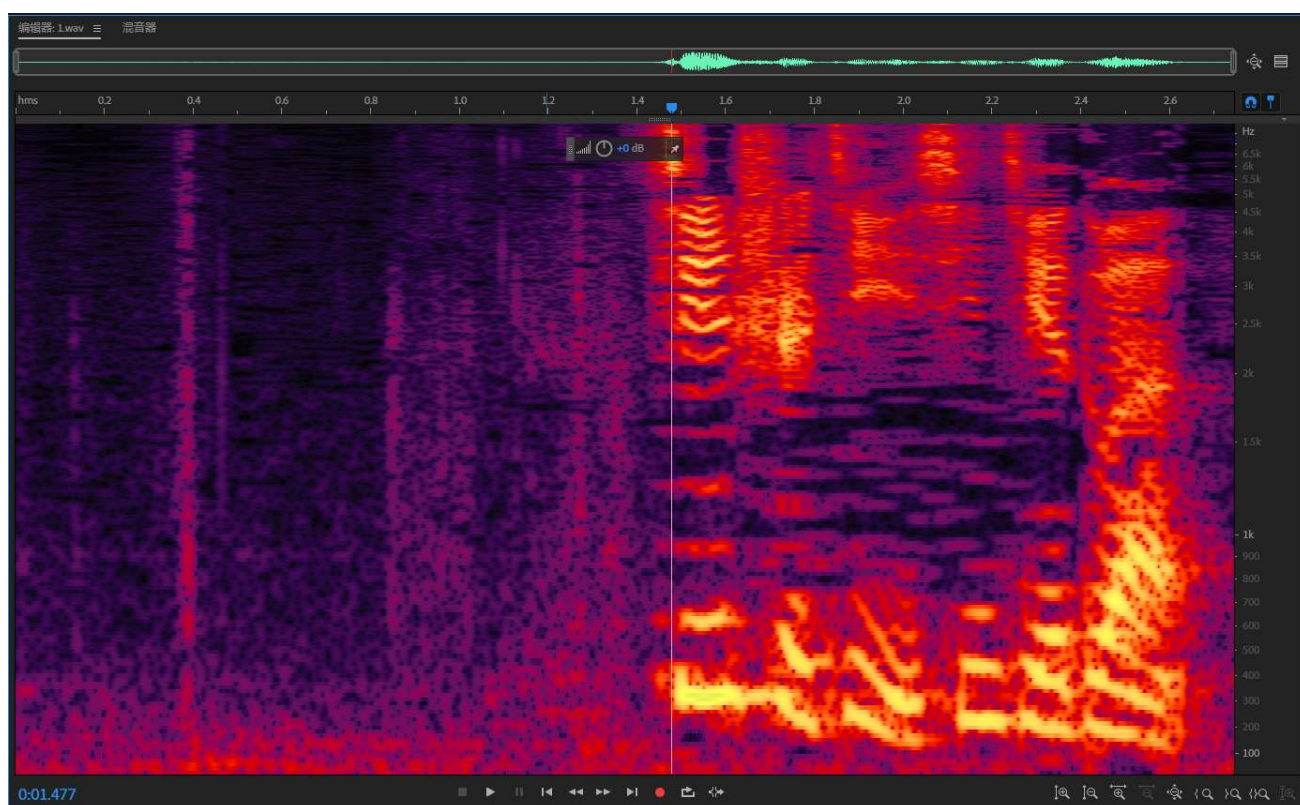
计算机科学与技术学院

一、语音编辑和处理工具的使用

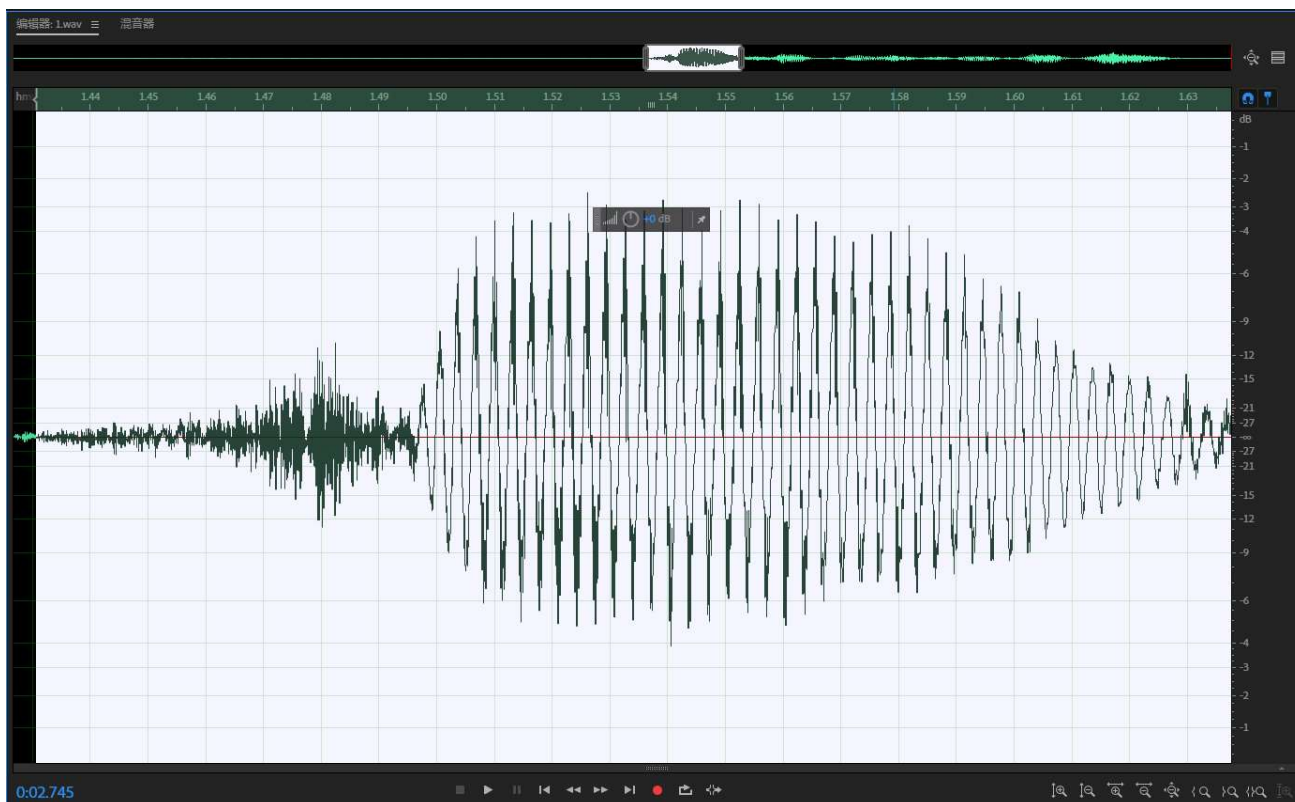
1.1 语音文件的时域波形截图



1.2 语音文件的语谱图截图



1.3 第一个音节的时域波形截图



1.4 语料格式

- 采样频率 = 16000Hz
- 量化比特数 = 16位
- 声道个数 = 1(单声道)

二、能量和过零率特征提取

以下所有均使用python开发

- python 3.7.0 32-bit
- macOS

2.1 给出特征提取算法，标明所采用的开发工具

对于过零率，我们由以下公式给出：

$$zcr = \frac{1}{N-1} \sum_{t=1}^{T-1} \mathbb{I}\{s_t s_{t+1} < 0\}$$

其中

$$\mathbb{I}\{A\} = \begin{cases} 1, & \text{if } A \text{ is true.} \\ 0, & \text{if } A \text{ is false.} \end{cases}$$

$$N = \text{length of frame window}$$

所以给出过零率的检测算法

```
def ZeroCR(waveData, frameSize):
    wave_len = len(waveData)
    frameNum = math.ceil(wave_len / frameSize)
    ZRresult = [] #最终结果
    f = []
    loop = 1
    result = 0
    for i in range(frameNum):
        l = loop + frameSize
        if l > wave_len:
            l = wave_len
            frameSize = l - loop
        for j in range(loop, l): # 平移窗口
            result = result + (waveData[j]*waveData[j-1] < 0)
        f.append(result)
        result = 0
        loop += frameSize

    for i in range(len(f)):
        ZRresult.append(f[i] / (frameSize - 1))
    return ZRresult
```

对于能量，我们由以下公式给出：

$$E = \sum_{n=1}^N x^2(n) \cdot w^2(n)$$

其中

$$w(n) = \frac{1}{2N}$$

```
def Energy(waveData, frameSize):
    wave_len = len(waveData)
    frameNum = math.ceil(wave_len / frameSize)
    ENresult = []
    loop = 0
    result = 0
    for i in range(frameNum):
        l = loop + frameSize
        if l > wave_len:
            l = wave_len
            frameSize = l - loop
        for j in range(loop, l):
            result = result + pow(waveData[j], 2) * pow(1.0 / (2 * frameSize), 2)
        ENresult.append(result)
        result = 0
        loop += frameSize
    return ENresult
```

三、端点检测算法

3.1 给出端点检测算法，标明所采用的开发工具

以下所有均使用python开发

- python 3.7.0 32-bit
- macOS

对于端点检测，使用了双门限进行判断。

在这里我使用了过零率和能量来作为门限来进行限制。为每个特征设置高低两个门限，为每一帧设置了三个状态。三个状态分别是Mute，Transition和Speech，进行标注之后筛选开始结束点取样。

首先我们设置四个门限，在两个特征均小于低门限时设置状态为Mute。当其中任何一个超过了低门限时设置状态为Transition。当两个状态均超过高门限时设置为Speech。

当从Speech状态跌回Transition时检查是否持续。如果没有就将Transition设Mute。如果Speech持续时间不超过5帧，判Speech为噪声，设为Mute。

这样可以得出一个包含帧状态的数组，然后进行计算得到开始结束的点，取出字节进行写文件。

状态判断分为两个阶段，一是直接判断。从数据直接进行判断获得状态而不进行逻辑筛查。二是逻辑筛查，判断是否是短时噪声以及是否是静音。通过两次判断将数组的所有数据洗成只有Mute和Speech的数据，方便设置起始点和终止点。

一次判断

```
def getState(zerodata, engdata, zero_low, zero_high, eng_low, eng_high):
    state = STATE.Mute
    if(zerodata < zero_low and engdata < eng_low):
        state = STATE.Mute
    if(zerodata >= zero_low or engdata >= eng_low):
        state = STATE.Transition
    if(zerodata >= zero_high and engdata >= eng_high):
        state = STATE.Speech
    if(state == STATE.Transition):
        if(zerodata >= zero_high or engdata >= eng_high):
            state == STATE.Speech
        elif(zerodata < zero_low and engdata < eng_low):
            state == STATE.Mute
    return state
```

二次判断

```
for i in range(len(zerodata)):
    state = getState(float(zerodata[i]), float(engdata[i]), zero_low, zero_high,
```

```

if(state == STATE.Speech):
    for j in range(1, 6):
        if i + j >= len(zerodata):
            s = len(zerodata)
        else:
            s = i + j
        state = getState(float(zerodata[s]), float(engdata[s]), zero_low, zer
        if state == STATE.Mute:
            break
        else:
            state = STATE.Speech
elif(state == STATE.Transition and i > 0 and result[i - 1] == STATE.Speech):
    state = STATE.Speech
else:
    state = STATE.Mute
print (str(i + 1) + ' ' + str(state) + '\n')
result.append(state)

```

两点筛查

```

begin = []
end = []
for i in range(len(result)):
    if i > 0 and result[i] == STATE.Speech:
        if result[i - 1] != STATE.Speech:
            begin.append(i * 256)
    if i > 0 and result[i] == STATE.Mute:
        if i > 1 and result[i - 1] != STATE.Mute and result[i - 2] == STATE.Speech:
            end.append(i * 256)

```

四、 计算检测正确率

4.1 “1.wav”语料去除静音后的时域波形截图



4.2 正确率

- 正确检出文件的个数: 10
- 正确率 = 100%

五、总结

5.1 请总结本次实验的收获

本次实验较为全面的了解了使用程序来对语音文件进行处理的全部过程，并且亲手编写算法进行运行。在实验中查询了大量的资料，对于语音处理有了大致的认识。对于双门限的理解更加的深刻。

5.2 请给出对本次实验内容的建议

希望能够加入更多的实验内容