

信号处理原理实验报告

李晨昊 2017011466

2019

目录

1	exp1	1
2	exp2	2
3	exp3	3

1 exp1

函数 `dtmf_fft` 实现了基于 FFT 的 DTMF 识别, 函数 `dtmf_goertzel` 实现了基于 Goertzel 算法的 DTMF 识别。

`dtmf_fft` 的大致思路为对输入信号进行 FFT, 然后在分别在行频率和列频率对应取值中找最大者, 并认为输入信号就对应于这一行和这一列。

`dtmf_goertzel` 的大致思路为模拟 IIR 滤波器的运行。在每个频率点上计算系数 $f = 2 \cos(\frac{2\pi k}{N})$, 以之为参数运行滤波器 $y(n) = x(n) + f y(n-1) - y(n-2)$, 然后分别选择行频率和列频率对应的功率最大的频率。经测试直接按照这个逻辑去写, 遍历每个频率, 结果会慢于 `dtmf_fft`, 所以我采用了向量化来加速计算, 逻辑上相当于同步运行 8 个独立的滤波器, 这样会快于 `dtmf_fft`。

实际使用这两个函数的时候需要把音频先分段再输入, 而且为了尽量避免噪音的影响, 这两个函数中我都设定只有功率超过一定值的时候才认为输入信号的确对应一个按键。

我分别使用了一段合成的音频和一段实际录制的音频进行测试。合成的音频内容为 0123456789ABCD*#, 实际录制的音频内容为 18*740#5#4*6。实验结果如下:

```
dtmf_goertzel on generated input
Elapsed time is 0.094947 seconds.
00000000001111111122222222223333333333444444444455555555556666666666777777...
```

```

dtmf_fft on generated input
Elapsed time is 0.160979 seconds.
00000000001111111122222222223333333333444444444455555555556666666666777777...
dtmf_goertzel on real input
Elapsed time is 0.039187 seconds.
1188**77744400##55##445***664
dtmf_fft on real input
Elapsed time is 0.049393 seconds.
1188**774400##55##44**66

```

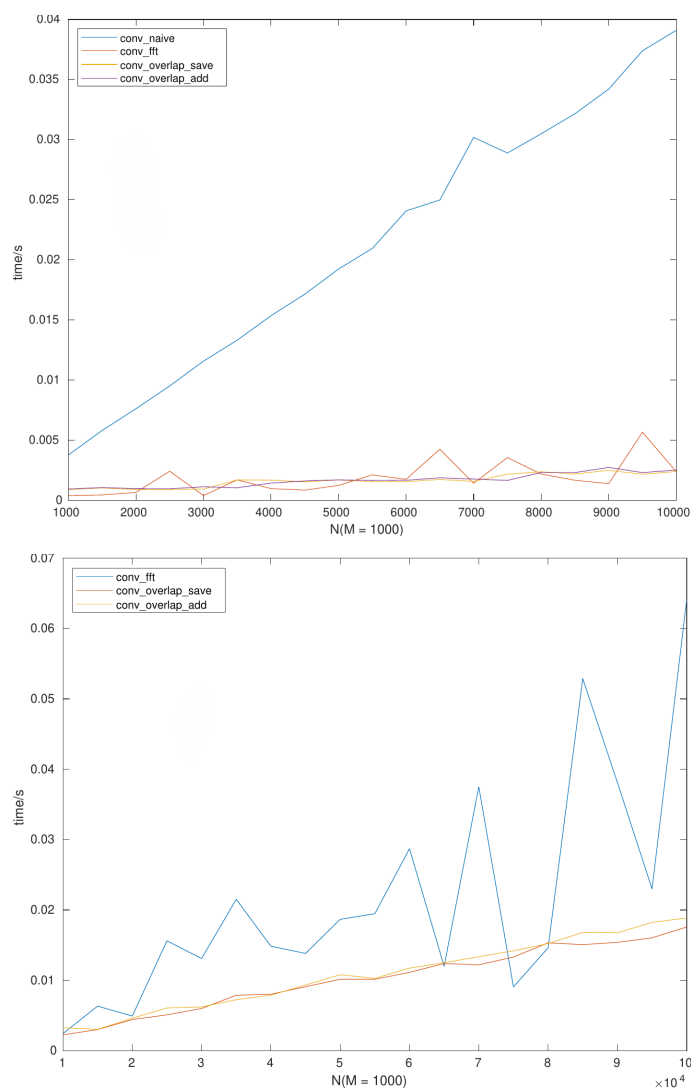
可见向量化加速之后 `dtmf_goertzel` 的确略快于 `dtmf_fft`，而且在合成的音频上能做到完全准确，不过在实际录制的音频上的精度稍差一些。

2 exp2

- 函数 `conv_naive` 实现了基于原始卷积公式的卷积计算
- 函数 `conv_fft` 实现了基于 FFT 的卷积计算
- 函数 `conv_overlap_save` 实现了基于 Overlap-Save 算法的卷积计算
- 函数 `conv_overlap_add` 实现了基于 Overlap-Add 算法的卷积计算

`conv_overlap_save` 和 `conv_overlap_add` 中，都随意地选择了块大小 = $4M$ 。这个参数对性能有很大的映像，如果是更实际的应用，则应该更认真的选择参数，不过我们这个简单的课程作业就没有必要考虑太多了。经测试块大小的选择不会影响正确性，证明程序的其他部分还是足够鲁棒的。

我进行了两组测试，测试的输入参数都在图中表示了。第二组中没有加入 `conv_naive`，主要是因为它太慢了，会干扰其他几个的对比。从结果上来看，`conv_overlap_save` 和 `conv_overlap_add` 的速度基本一致，当两个输入信号的长度相差悬殊的时候，性能基本上都好于 `conv_fft`。如果认真选择块大小的话，也许效果会更好。



3 exp3

`exp3.m` 中实现了三个信号的频分复用。过程与实验指导描述的基本一样，先对信号进行补零上采样，再转换到频域，再将三个信号各自保留一个周期，填充到待传输信号的频域中，最后可以从回复出原始的时域信号。

这三个音频是网络上下载的语音包，都附在提交的文件当中了，虽然其长度超过了 6 秒，但是它们在 6 秒基本都没有声音了，所以代码中就按均为 6 秒来处理了。实验中涉及到的各个信号的图像如下，可以看出信息几乎没有任何损失，人耳听起来也几乎没有任何区别。

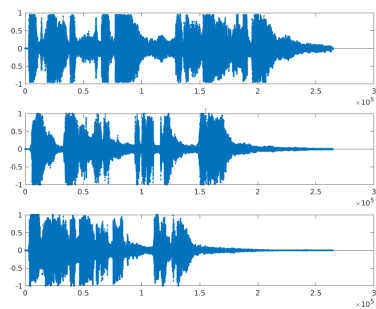


图 1: 原始信号时域

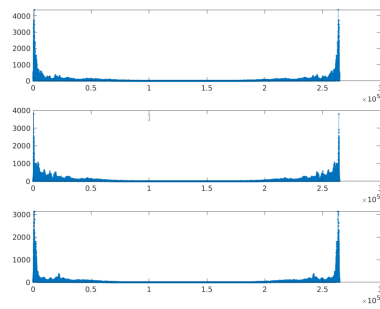


图 2: 原始信号频域

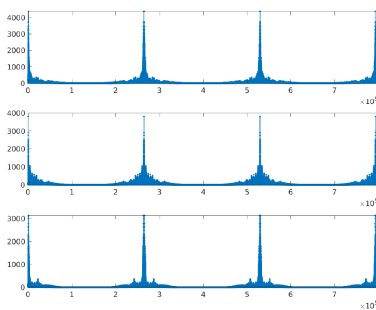


图 3: 上采样后信号时域

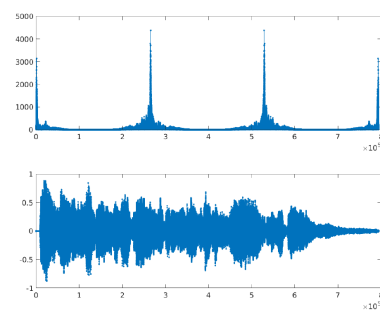


图 4: 传输信号时域和频域

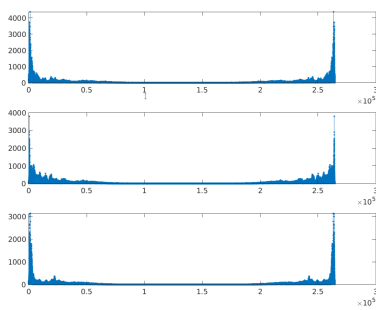


图 5: 恢复信号频域

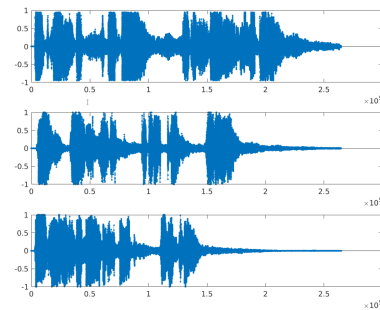


图 6: 恢复信号时域