

信号处理原理实验报告

2017011620 计 73 李家昊

2020 年 1 月 11 日

1 实验 1

本实验中实现了 Goertzel 算法识别 DTMF 音频信号，并在速度和精度上与 FFT 做了对比。

1.1 标准音频测试

在网上下载了按键 0-9 的 DTMF 音频，分别用 FFT 和 Goertzel 算法识别，测试结果如 Table 1所示。

Key	Length	Goertzel Result	Goertzel Time (s)	FFT Result	FFT Time (s)
0	4687	0	0.001912	0	0.002677
1	4049	1	0.001160	1	0.002206
2	3731	2	0.001244	2	0.002253
3	5254	3	0.001684	3	0.002409
4	5326	4	0.002075	4	0.002727
5	4523	5	0.001485	5	0.002248
6	5613	6	0.001662	6	0.002197
7	5071	7	0.001887	7	0.002594
8	4918	8	0.001498	8	0.002282
9	4658	9	0.001441	9	0.002234

Table 1: Goertzel 和 FFT 算法的标准单按键音频测试结果

可以看出，在精度上，两种算法的预测结果都非常准确，在速度上，Goertzel 比 FFT 更快一些，原因是其计算量更小，且序列长度并不大。

1.2 实际音频测试

接下来用手机拨打我的电话 13380831033，并用笔记本电脑将按键音频录制下来，其中伴随少量环境噪声，原始时域信号如 Figure 1所示。

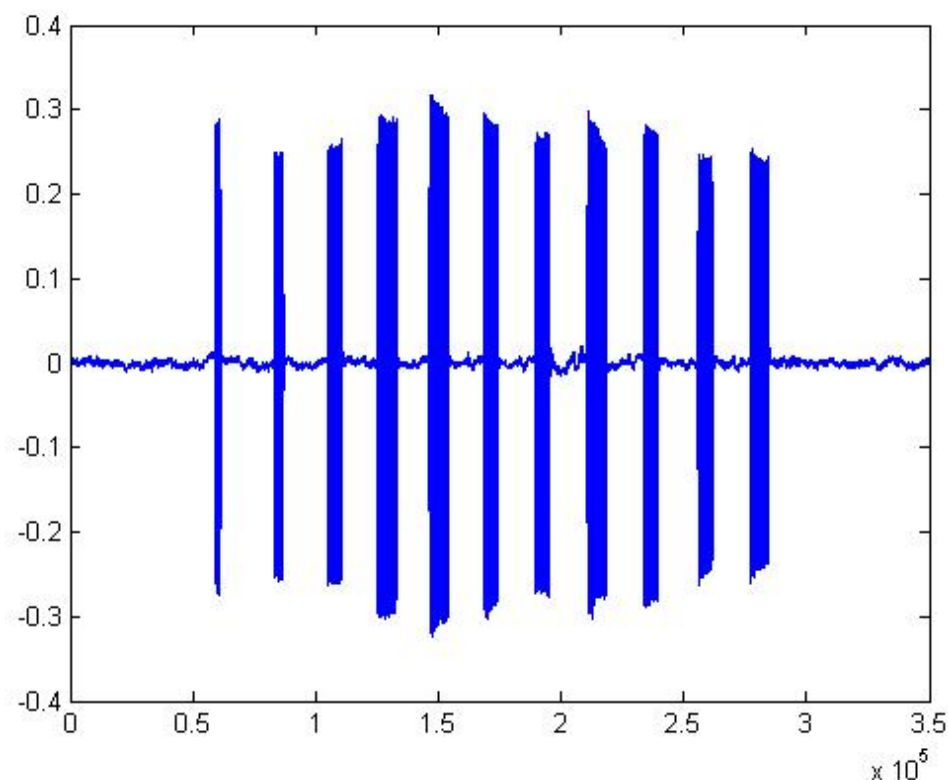


Figure 1: 实际音频的时域信号

从音频的第 57000 个采样点处开始，每 22000 个采样点为一个按键，共分成 11 个按键，取每个按键的前 5000 个采样点，分别采用 Goertzel 和 FFT 算法测试，结果如 Table 2所示。

Key	Goertzel Result	Goertzel Time (s)	FFT Result	FFT Time (s)
13380831033	13380831033	0.010394	13380831033	0.025543

Table 2: Goertzel 和 FFT 算法的实际多按键音频测试结果

可以看出，在实际情况下，两种算法都给出了精确的结果，但在速度上 Goertzel 仍然比 FFT 更快。

2 实验 2

本实验中实现了四种卷积算法，为评测它们的性能，令序列总长度 L 从 2 递增到 5000，令序列 x 的长度 L_x 在 $[1, L - 1]$ 中随机产生，序列 y 的长度为 $L_y = L - L_x$ ，序列 x, y 的内容均随机生成，用四种算法分别计算 $x * y$ ，统计所用的时间，测试结果如 Figure 2 所示。可以看出，直接按公式计算的效率明显低于其他三种算法，其他三种算法性能相当，FFT 的速度略低于 Overlap-Add 和 Overlap-Save，因为这两种算法在 L_x 和 L_y 相差较大时具有不错的加速效果。

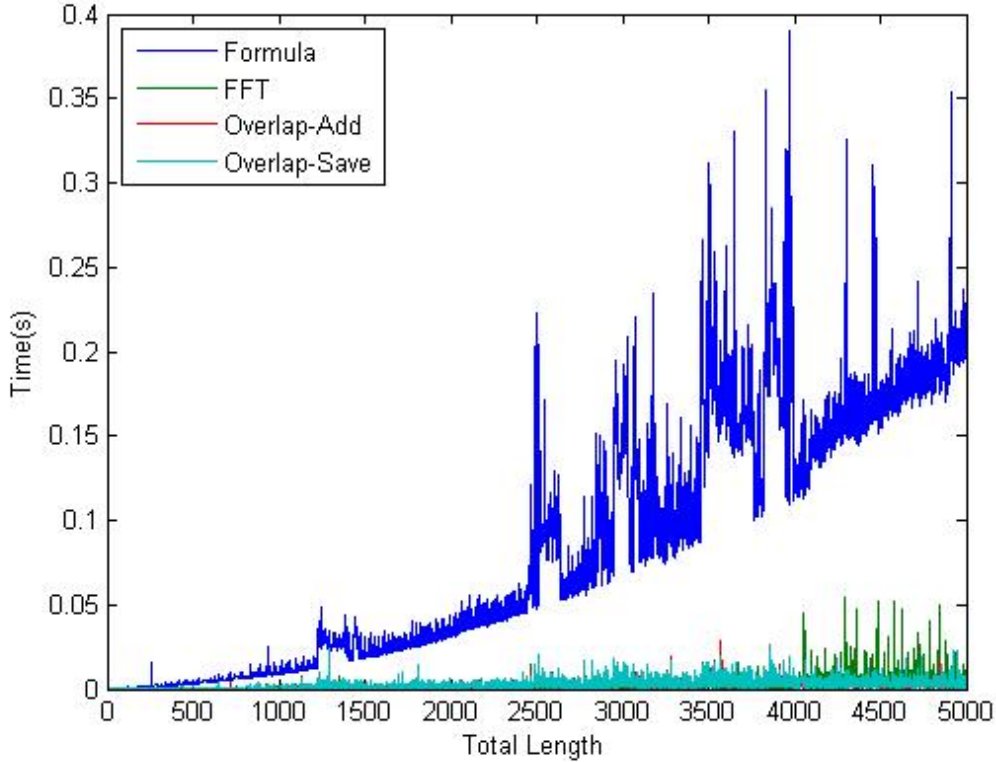


Figure 2: 四种算法的运行时间随序列总长度的变化曲线图

3 实验 3

本实验中模拟了频分复用（FDM）的编码和解码过程。设有 n 个长度为 L 的音频，如图 Figure 3 所示，在调试过程中，将每个音频上采样 n 倍，如图 Figure 4 所示，然后通过 FFT 得到频域信号，通过滤波器互不重叠地提取每个音频频谱中的一个周期，将这些频谱信号叠加起来，就得到了调制信号，如图 Figure 5 所示。在解调过程中，用滤波器将对应频段的调制信号提取出来，再进行 IFFT 即可得到时域信号，如图 Figure 6 所示。

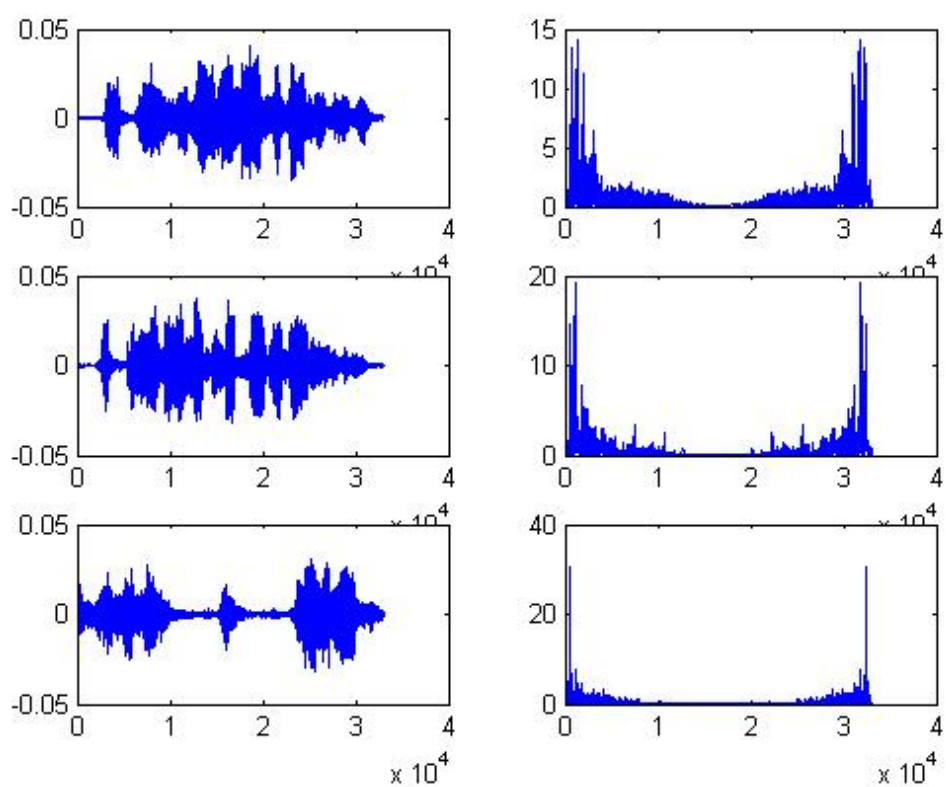


Figure 3: 原始的时域信号（左）及频域信号（右）

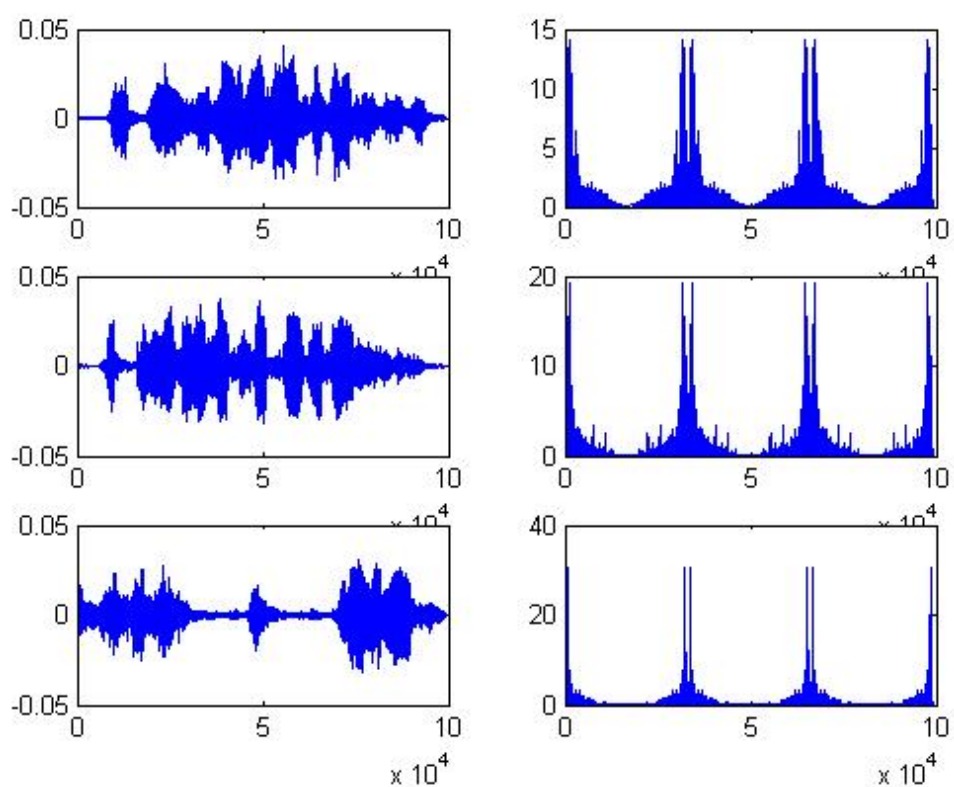


Figure 4: 上采样后的时域信号（左）及频域信号（右）

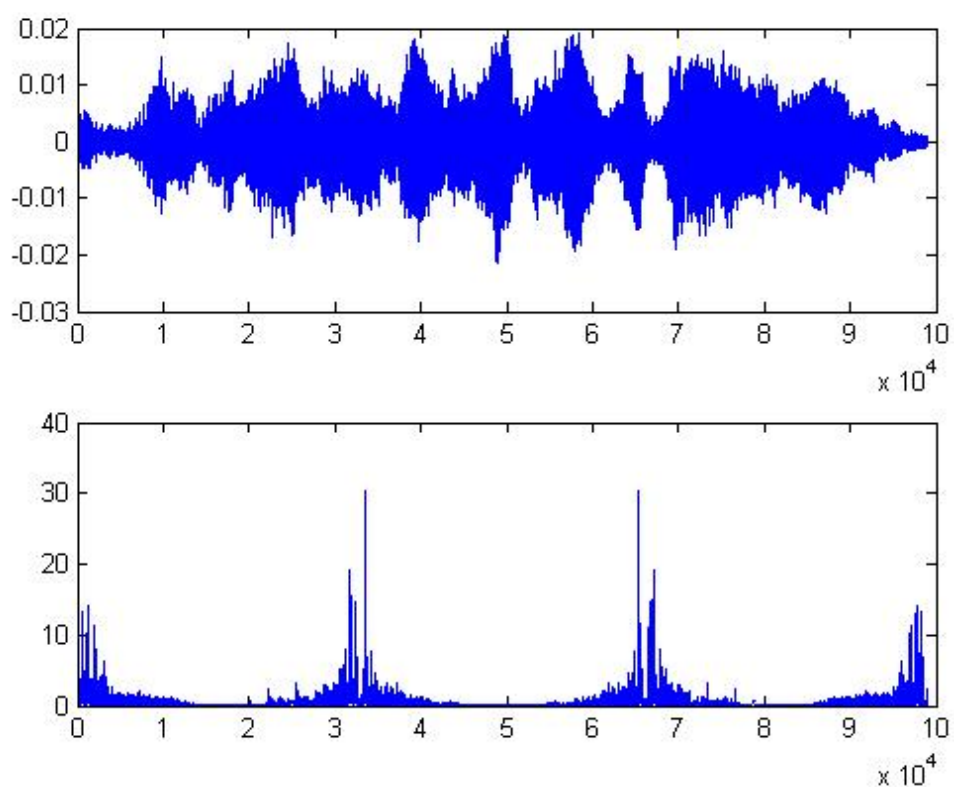


Figure 5: 调制后得到的时域信号（上）及频域信号（下）

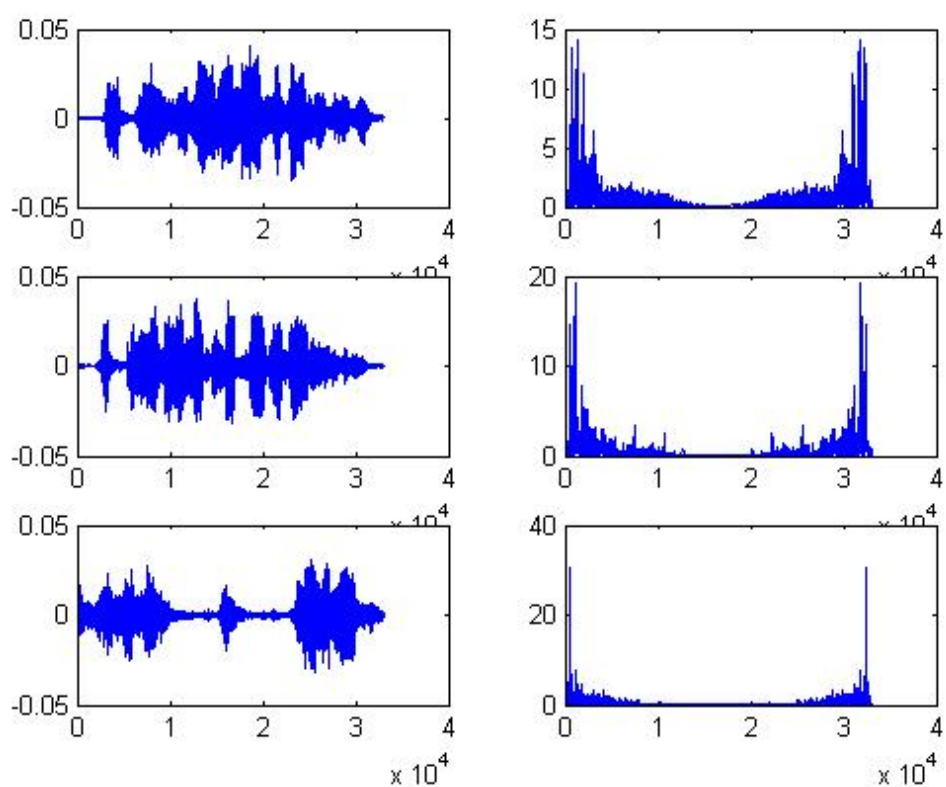


Figure 6: 解调后得到的时域信号（左）及频域信号（右）