二组课程设计

作者:[鲁国锐、钱军、马健康、马常昊、王志轶、王清扬、徐润芯]

指导教师：[李光亮]

课程名称：[信号与系统]

分工：

图像组：鲁国锐、钱军、马健康、马常昊

音频组：王志轶、王清扬、徐润芯

2019 年 12 月 12 日

目录

傅里叶高通滤波器

一、 任务描述

对图像进行傅里叶变换，再通过得到的频谱图对其进行高通滤波。

二、 问题分析

简单起见，本次任务中我们只讨论灰度图。

首先我们需要明确一下什么是图像的频率。对于灰度图而言，其频率表征的是灰度变化的剧烈程度。也就是说，灰度跳变大的地方，如边缘，代表高频部分；而灰度变化平缓地地方，如大片的蓝天，代表低频部分。

通过对图像频率的分析我们不难发现，对图像进行高通滤波，就是要突出图像中变化剧烈的部分，而使变化平缓地部分更加平缓，甚至不变。

我们还可以预测一下结果，当我们对图像进行了高通滤波后，图像的所有边缘会被提取出来，而其它大片变化平缓的背景会变得更暗（因为少了大量低频分量的能量）。稍后我们会在实验结果中验证这一猜想。

接下来就是如何操作了。由于灰度图像是二维的，相当于有两个时间轴，所以对应的我们也要用二维的傅里叶变换。这里列出可能用到的公式（如图 1 所示），并不对其作深入的、数学方面的讨论。我们只需要知道在用完傅里叶变换之后便能得到其频谱图，而在 MATLAB 中我们可以直接调用 fft2 函数完成变换。**但需要注意的一点是，频谱图中的坐标并不与原图对应，傅里叶变换后频谱图上的每一点值与原图中所有点的值都有关系。**

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-i2\pi(ux/M+vy/N)}$$

图 1 二维离散傅里叶变换公式

同时为了操作的方便，我们还要调用 fftshift 函数，把低频分量移到频谱图的中心。关于这个函数的细节我们通过一张图[1]来说明（如图 2 所示），也不做过多解释。

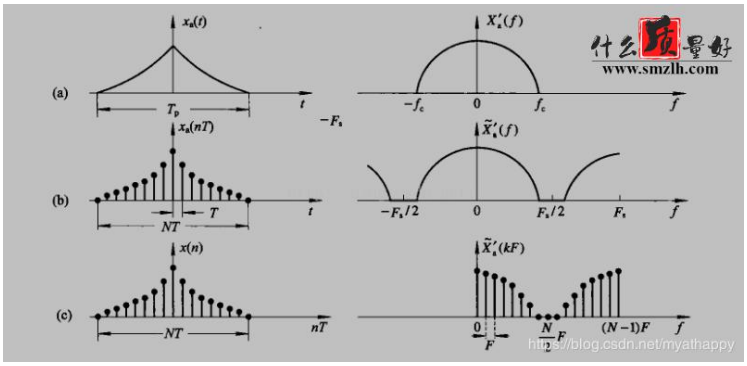


图 2 fftshift 图解

通过了 fftshift 函数之后的频谱图从中心向四周频率逐渐降低。此时我们只需要在其中心画一个圈，圈内的为低频分量，全部置为 0；圈外的为高频分量，不做改变。最后再进行

[1] 图像来自于网站：https://blog.csdn.net/myathappy/article/details/51344618

一次反变换，便得到了高通滤波后的图像。

三、 解决方案

利用imread函数读取电脑中的一张照片，再用rgb2gray函数将三维彩色图片转成二维灰色图片。通过imshow函数显示变换后的灰色图像。然后对图像进行傅里叶变换，展示经过傅里叶变换后的频谱图。然后利用size函数算出图片像素的行数a和列数b，利用round函数找到图片的中心点坐标（a_0,b_0)。设置相应的参数d(图片中任一点到中点距离的参考值)，利用循环，找到每一点到中点的距离，如果大于参数d，置变量h=1；反之h=0（h为0，表示该点频率能量置零;h为1，表示该点频率能量不变），那么原图每点的频谱s(i,j)=h*s(i,j)。最后利用imshow函数显示低通滤波后的图片。

四、 流程框图

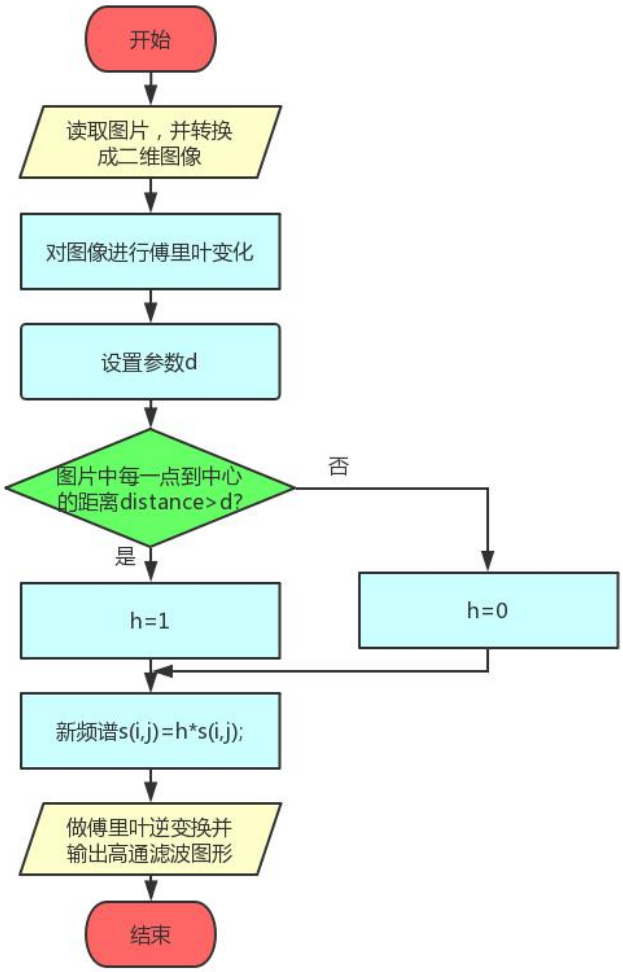


图 3 高通滤波流程图

五、 编程实现

```
I=imread('C:\Users\Asus-\Desktop\signal_and_system\img1.jpg');
I=rgb2gray(I);
figure(1),imshow(I);
title('原图像');
s=fftshift(fft2(double(I)));
figure(2),imshow(abs(s),[]);
title('图像傅里叶变换所得频谱');
figure(3),imshow(log(abs(s)),[]);
title('图像傅里叶变换取对数所得频谱');
[a,b]=size(s);
a0=round(a/2);
b0=round(b/2);
d=80;



for i=1:a
      for j=1:b
          distance=sqrt((i-a0)^2+(j-b0)^2);
          if distance<=d
              h=0;
          else
              h=1;
          end
          s(i,j)=h*s(i,j);
      end
end
s=uint8(real(ifft2(ifftshift(s))));
figure(4),imshow(s);
title('高通滤波所得图像');
figure(5),imshow(s+I);
title('高通滤波所得高频增强图像');
```

六、 结果展示

图 4 原图

图 5 灰度图像

图 6 傅里叶变换所得频谱图

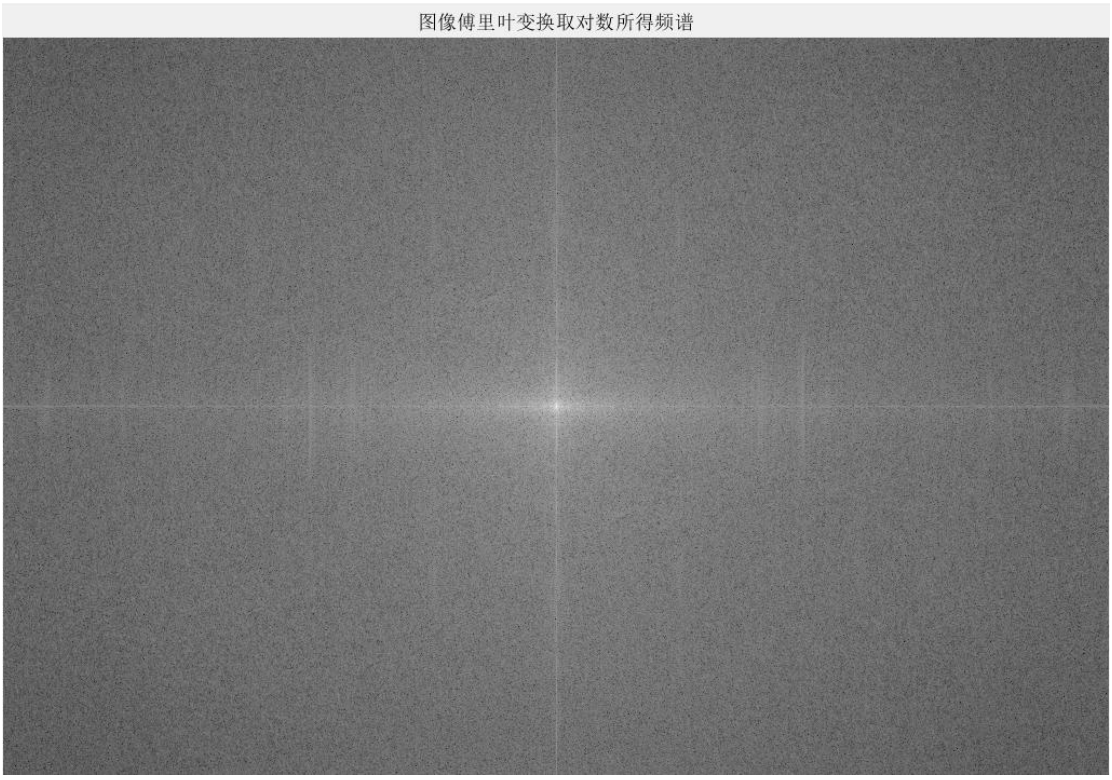


图 7 傅里叶变换取对数所得频谱图

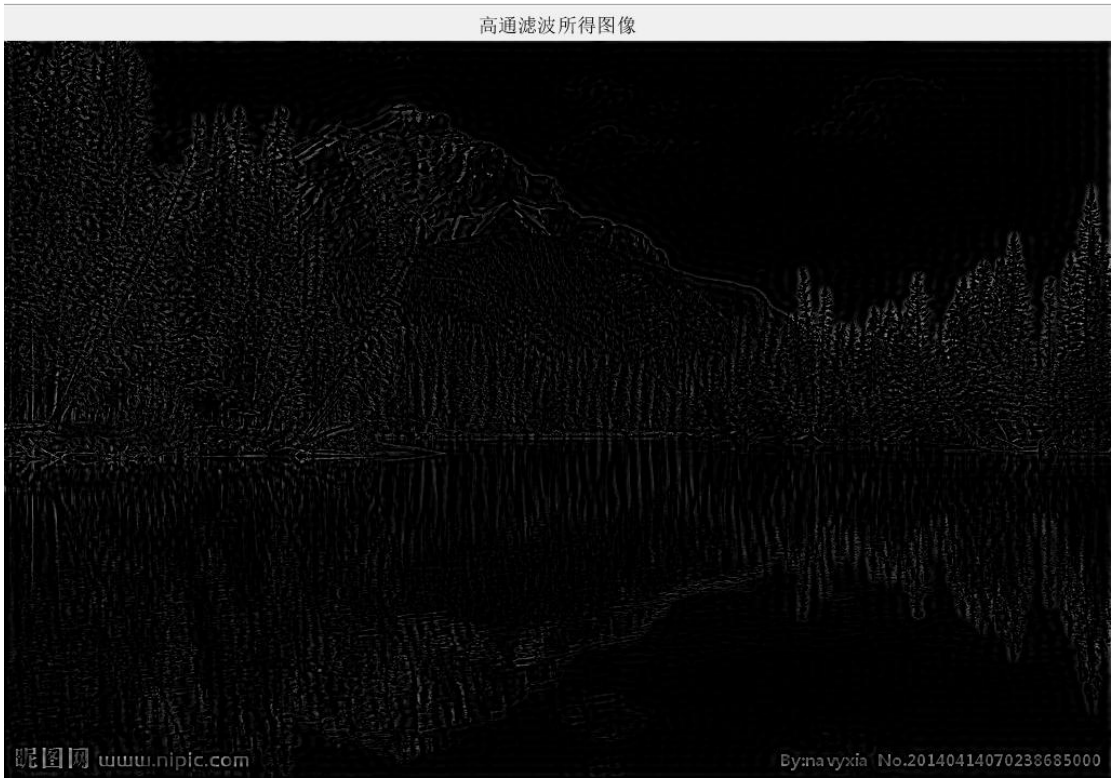


图 8 高通滤波所得图像

七、 分析总结

从实验结果可以看出，高通滤波的结果基本与我们预期相符合。

我们也发现，在用 imshow(abs(s),[])展示时，会对矩阵进行一个预处理：

```
b=(a-min(a(:)))./(max(a(:))-min(a(:)))*255;
imshow(uint8(b));
```

又由于频谱图每一点的模值差别很大，这样处理完之后绝大多数值都几乎为 0，展示出来除了中间的一个亮点其余都完全是黑色，看不出差别。所以我们对其做了一个取对数的操作，这样可以更明显地看出频率分布。

在频谱图中越亮的地方代表对应的频率能量越强，所以我们也可以看出，一副图像中的绝大部分能量都集中在了低频部分。

拉普拉斯滤波

一、 任务描述

用拉普拉斯滤波的方法处理图像。

二、 问题分析

本次任务中我们仍以灰度图为例。

拉普拉斯滤波也可用于图像锐化。它的原理很简单：对每一个像素，把它上下左右的像素值与之相减，得到的差取绝对值再相加，最后将和放入该像素对应的位置。这样做可以起到提取边缘的效果，因为边缘即像素值变化大的地方，那么最后得到的差值之和也会更大。

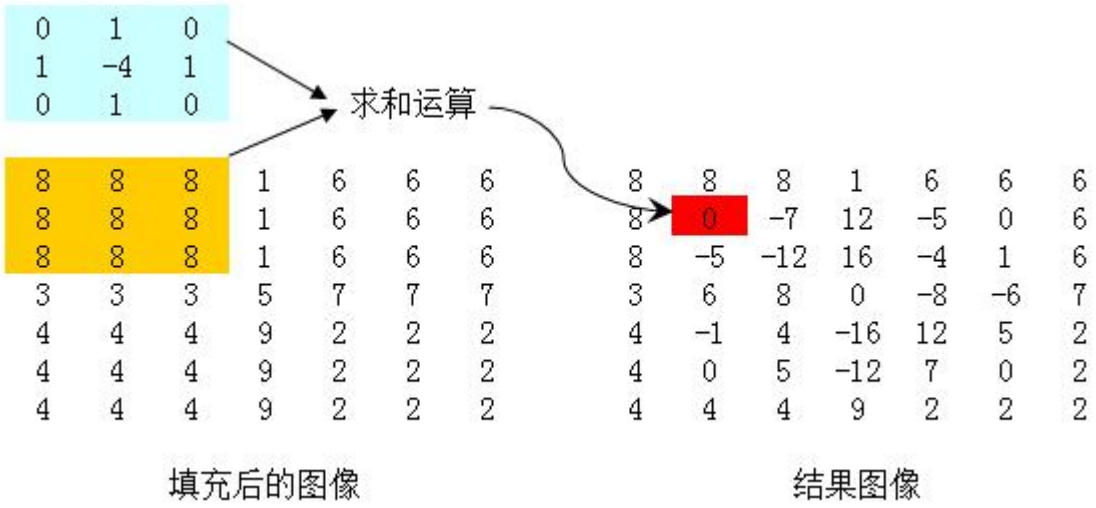


图 9 拉普拉斯滤波图解[2]

在次任务中，为了使提取效果更明显，我们还加上了四个角上的像素值与中心像素值的差。

三、 解决方案

在处理一张图片之前，首先给出图片文件所在的路径，由程序读取出原图片文件的数据信息。将彩色图片灰度图以后再开始对其每个像素点进行处理。

具体方式为：每个像素点的数值变为周围八个像素点的数值之和与八倍原像素点的数据之差。由点（2，2）处理至点（x-1，y-1）即完成所有操作处理。最后将我们所得处理后的图形输出即可。

[2] 图片来自网络：https://blog.csdn.net/scottly1/article/details/44408343

四、 流程框图

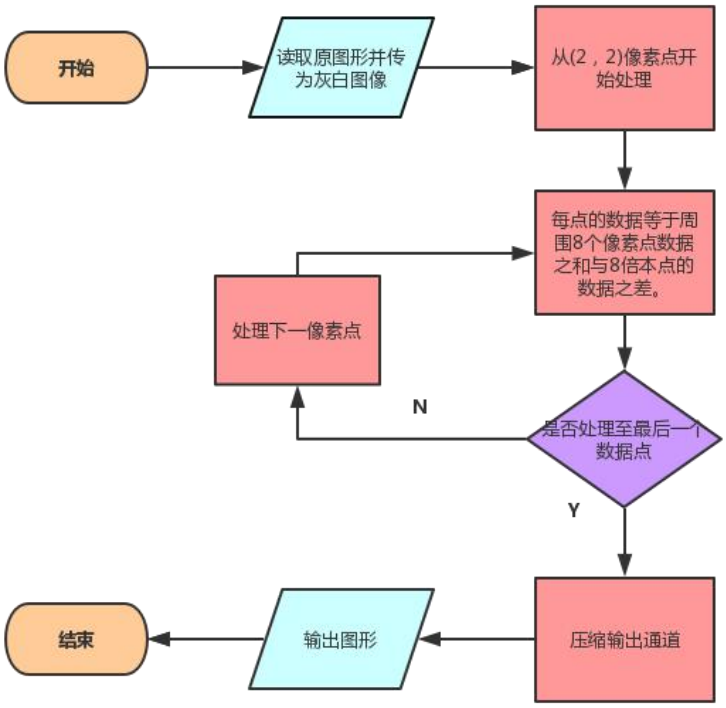


图 10 拉普拉斯滤波流程图

五、 编程实现

```
%使用拉普拉斯算子实现图像的边缘提取
I=imread('C:\Users\Asus-\Desktop\signal_and_system\timg.jpg');      % 读取图像
I=rgb2gray(I);
I=im2double(I);
[M,N]=size(I);
B=zeros(size(I));

for x=2:M-1
    for y=2:N-1
        B(x,y)=I(x-1,y-1) + I(x+1,y-1) + I(x-1,y+1) + I(x+1,y+1) +
I(x+1,y)+I(x-1,y)+I(x,y+1)+I(x,y-1)-8*I(x,y);
    end
```

```
end


I=im2uint8(I);
B=im2uint8(B);
figure(1);
subplot(121);imshow(I);
subplot(122);imshow(B);
```

六、 结果展示

图 11 原图

图 12 拉普拉斯滤波后的图像

七、 分析总结

从实验结果可以看出，拉普拉斯滤波基本上能起到提取边缘的作用。但同时也会对图像中的噪声进行加强。可先对图像进行低通滤波，在用拉普拉斯滤波进行锐化。

音频信噪声降噪处理

一、 任务描述

利用所学知识对一段音频信号进行加上白噪声（噪声），并且再进行降噪处理，从中了解声音信号的存储条件，了解信号的频域和时域特性，并进行分析。

二、 问题分析

本部分主要是基于 MATLAB 的语音信号的处理，首先我们想利用函数对语音信号的进行采样，将信号存储在数据中。然后利用 MATLAB 对其进行时域和频域的分析，例如快速傅里叶变换等，画出信号的幅频特性和时域波形图，通过研究各个信号在频域里的特性，结合各个频率分量的特点和我们想要的结果，对信号进行处理，例如滤波操作。

三、 解决方案

（1）高斯白噪声的获取

高斯白噪声，如果一个噪声，它的瞬时值服从高斯分布，而它的功率谱密度又是均匀分布的，则称它为高斯白噪声。

高斯白噪声是工程应用中及其常见的噪声干扰，例如收音机等各个通信设备在接收发送信号时由于电磁产生的干扰。高斯白噪声通常各个频率分量都存在，而且在时域上的强度基本相等（如下图）：

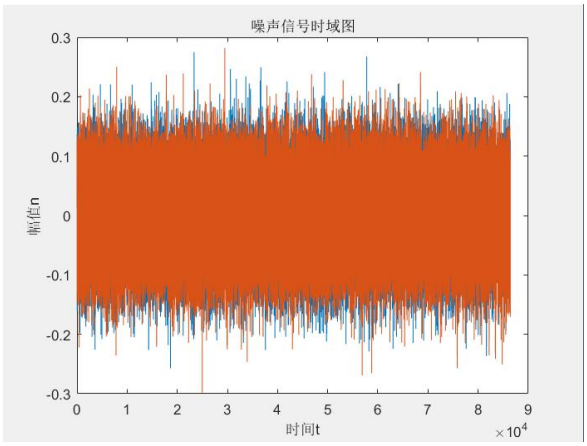


图 13 高斯白噪声时域图

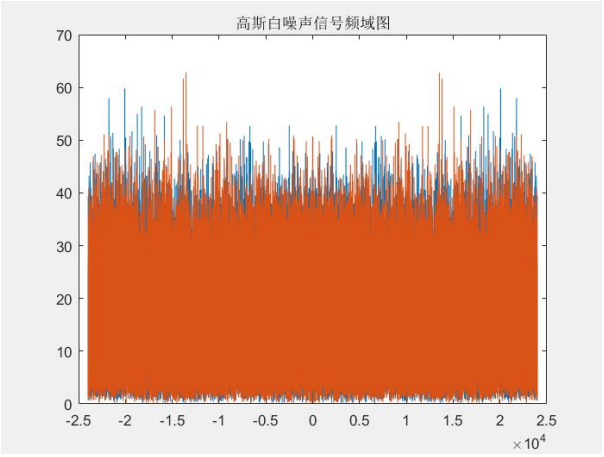


图 14 高斯白噪声频域图

这种信号的产生我在 MATLAB 中是用随机函数 randn()写的，通过 sound 函数播放声音可以播放其声音，经过测试可以听见明显的沙沙声音，我们不妨与一个语音信号进行叠加来模拟我们生活中高斯白噪声。

（2）采样信号的获取

这里语音信号的取样与于某软件里面的比较著名一段音频。但由于手机获取问题，还需要在电脑上利用特定软件进行其频域，时域特性如下图。

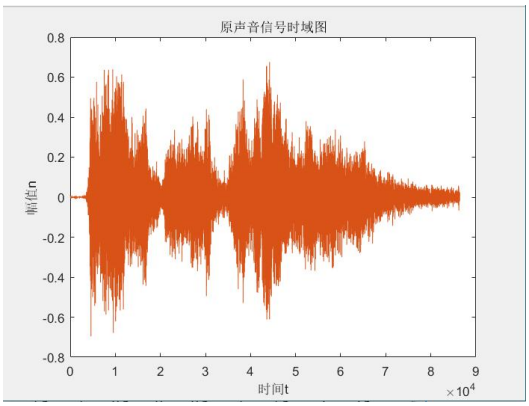


图 15 音频信号时域图

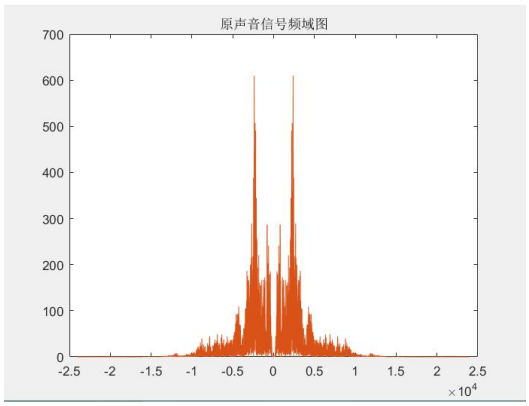


图 16 音频信号频域图

其采样函数应用 voiceread 函数，双声道音频读入存储在一个二维数组里。因此，声音的叠加是两个二维数组（噪声与音频）的叠加。以下是叠加后的信号频域图:

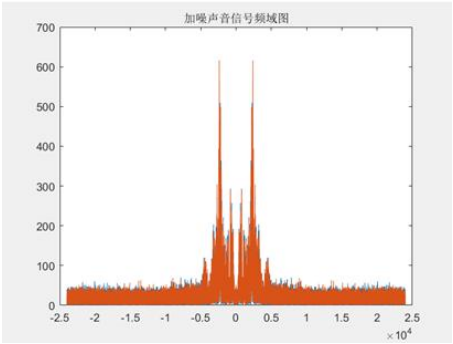


图 17 混合信号频域图

可也以看到，经过叠加之后，声音频域里高频信号明显强度增大，因此可以通过低通滤波器对高频的成分进行滤波，从而大大减少高频成分的干扰，但需要注意的是，低通滤波器只是仅仅对噪声信号的高频成分进行滤波，而低频信号中的成分由于与原信号混合在一起，所以无法滤波。但是我有一种专门的解决办法，**就是将所有频率的信号都减去高频信号（平均）的幅度**即：

$$A_f' = A_f - A_{fmax}$$

这样可以实现一种滤波方式，但是用 MATLAB 实现起来相对困难，尤其是广义的高频范围难以把握，因此我并没有对其进行处理，**而是采用相对简单的低通滤波器进行降噪处理。**关于低通滤波器，我采用的是一种 IIR 的巴特沃斯数字滤波器，这种滤波器实现起来相对简单，而且在 MATLAB 中有现成的函数可以使用。如利用 buttord 函数获取信号所需滤波器的阶数等参数，在使用 butter 设计出滤波器，因此相对简单。

以下是巴特沃斯滤波器的幅频特性曲线。

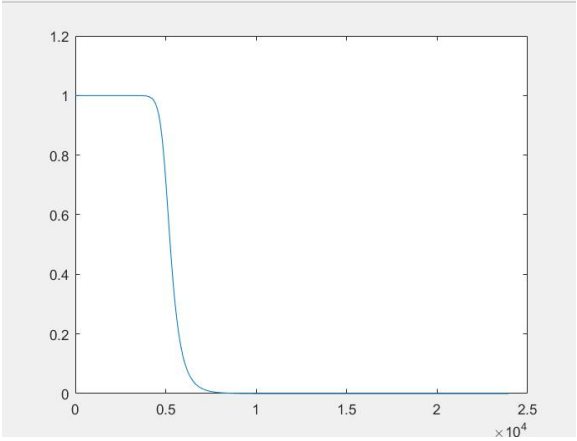


图 18 巴特沃斯滤波器特性曲线

由此可以看出来，当频率大于 5Khz 的信号会被滤掉，而小于 5Khz 的将会被保留。因此这是一个低通滤波，可以将将混合好的带有噪声的音频信号进行滤波，从而大大减小噪声的干扰，但是并不能完全消除这种噪声。

而这种滤波器对纯高频的噪声的降噪十分明显。例如以下这种噪声，纯高频的噪声，例如以下这种纯正弦波的噪声，常见于音响，喇叭等需要功放上面。低通滤波器便可以完全的将高频信号滤去。

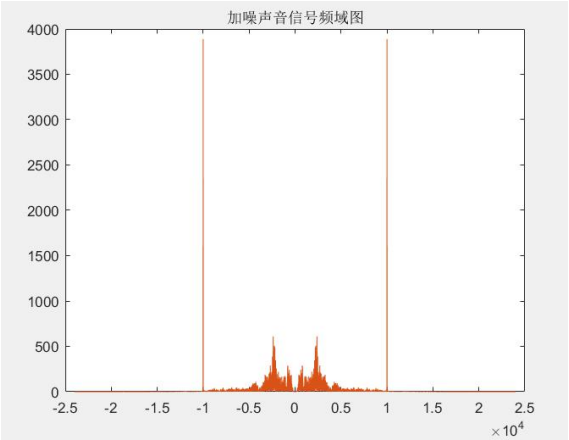
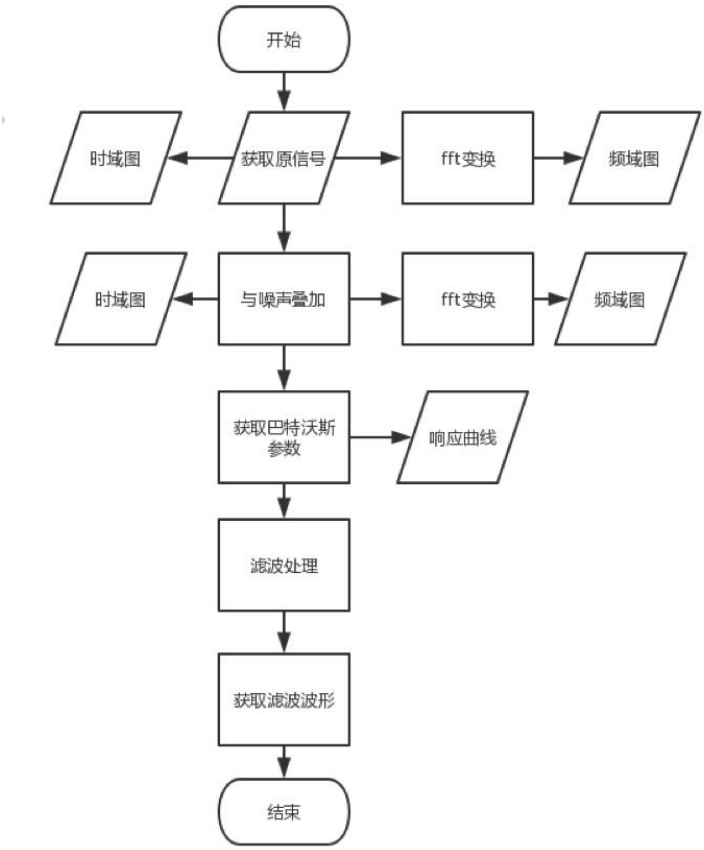


图 19 一种常见的高频信号噪声干扰

四、 流程框图



五、 编程实现

```
[x1,fs] =
audioread('C:\Users\wangzhiyi\Desktop\test_data0.wav')
;
%sound(x1,fs);

figure(1);
plot(x1);
title('原声音信号时域图');

xlabel('时间t ');

ylabel('幅值n ');


figure(2);
y1 = fft(x1);% 对音频信号进行快速傅里叶变换
y1 = fftshift(y1);
```

```
derta_fs = fs/length(x1);
plot([-fs/2:derta_fs:fs/2-derta_fs],abs(y1));
title('原信号声音频域图');
le = length(x1);

%%%%%%%产生高斯白噪声
noise1 = 0.06*randn(le,2);

%%%%%%%产生高频信号for a = 1:le;
    noise2(a,1) = 0.1*sin(2*pi*10000*(1/fs)*a);
    noise2(a,2) = 0.1*sin(2*pi*10000*(1/fs)*a);
   end;

%s = x1 + noise1;% 选择产生的噪音是高频还是白噪声s = x1 +
noise2;

%sound(s,fs);

figure(3);
plot(noise1);
title('噪声信号时域图');

xlabel('时间t ');

ylabel('幅值n ');

figure(7);
ns = fft(noise1);
ns = fftshift(ns);
derta_fs = fs/length(x1);
plot([-fs/2:derta_fs:fs/2-derta_fs],abs(ns));
title('高斯白噪声信号频域图');


figure(4);
y2 = fft(s);
y2 = fftshift(y2);
plot([-fs/2:derta_fs:fs/2-derta_fs],abs(y2));
```

```
title('加噪信号频域图');


%%%%%%%%巴特沃斯滤波器
rp = 2;
rs = 80;
Ft = 8000;
Fp = 400;
Fs = 1000;   %设置滤波器截止频率
wp = 2*pi*Fp/Ft;
ws = 2*pi*Fs/Ft;
[n,wn] = buttord(wp,ws,rp,rs,'s'); 得到滤波器的特性
[b,a] = butter(n,wn,'s');% 生成滤波器
[bz,az] = bilinear(b,a,0.5);

figure(5);% 绘制滤波器的特性曲线
[h,w] = freqz(bz,az);
title('低通滤波器幅频特性');
plot(w*fs/(2*pi),abs(h));

z = filter(bz,az,s);% 进行滤波操作
%sound(z,fs);

figure(6);
y3 = fft(z);
y3 = fftshift(y3);
plot([-fs/2:derta_fs:fs/2-derta_fs],abs(y3));
title('降噪噪声音信号频域图');
audiowrite('C:\Users\wangzhiyi\Desktop\test_data0_a.w
av',z,fs);
```

六、 结果展示

以下是经过滤波后的声音的频域图。其详细的文件在附件里。

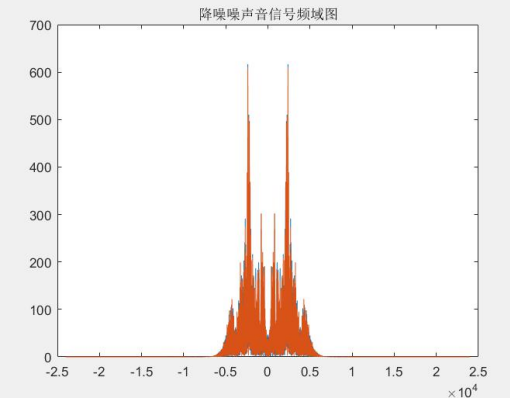


图 20 降噪后声音的频域图（高斯噪声）

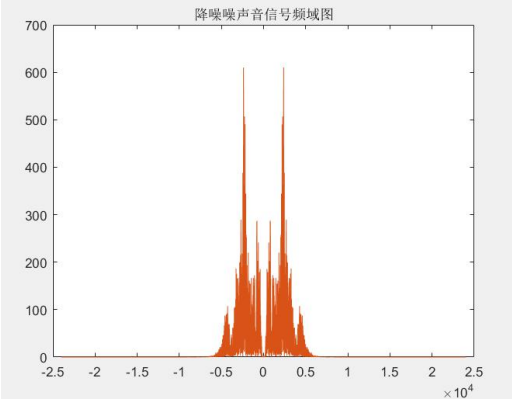


图 21 降噪后声音的频域图（高频信号）

七、 分析总结

从频谱上分析，两者并无差别，但是效果上，由于白噪声在各个频段都有，因此降噪效果并不是很明显，但是，高频信号相比较滤波更加完美，而且还有一种朦胧的感觉。

两者在听觉上可以明显感觉到失真，这都是因为高频信息的缺失导致，以牺牲音质为代价换来降噪的效果。但是有的场合，例如低音炮，变声器等等却要用到这种失真。

并且，由低通滤波器，我们还可以想到更多的降噪方法：

耳机的低频底噪：高通滤波器。

50hz 交流电干扰：带阻滤波器，等等。

音频信号的提取

一、 任务描述

根据傅里叶变换从多种混合音频中（不同频率）提取另外的一种波形。

二、 问题分析

这个问题与上个问题类似，都是对信号进行滤波处理，因此我们可以找到一种用于音频分离的思路，比如人的声音在大部分在几百赫兹，但是我们人耳的听觉范围往往比这个范围大的多。通过对信号进行傅里叶变换，进行滤波处理，可从不同的音频中得到有用的信息。

三、 解决方案

同上个问题，我们可以利用滤波器进行人与高频信号的分离，下面一张图是我在某著名歌曲中录制的一段音频。其中混杂着蛐蛐的叫声，以及人的声音。详见下图

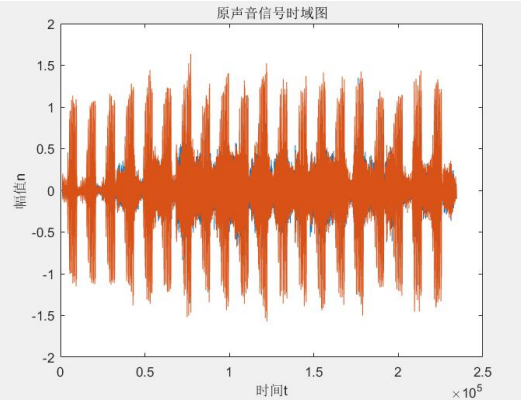


图 22 混合声音时域图

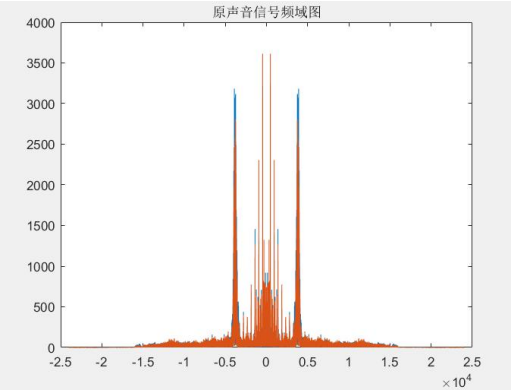
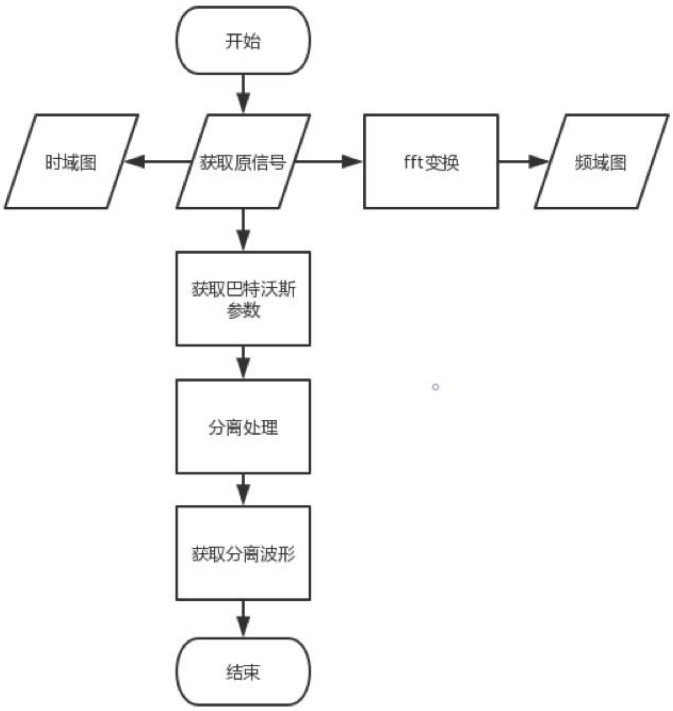


图 23 混合声音的频域图

右图中的高频部分的尖端信号就是蛐蛐的叫声，通过一个低通滤波器便可以将人的低频声音与之分离。滤波器的特性上题中有详细说明。

四、 流程框图



五、 编程实现

```
[x1,fs] =
audioread('C:\Users\wangzhiyi\Desktop\test_data2.wav')
;
%sound(x1,fs);

figure(1);
plot(x1);
title('混合信号时域图');

xlabel('时间');

ylabel('幅值');


figure(2);
y1 = fft(x1);
y1 = fftshift(y1);
derta_fs = fs/length(x1);
plot([-fs/2:derta_fs:fs/2-derta_fs],abs(y1));
title('原声音信号频域图');


rp = 2;
rs = 80;
Ft = 23000;
Fp = 600;
Fs = 1000;
wp = 2*pi*Fp/Ft;
ws = 2*pi*Fs/Ft;
[n,wn] = buttord(wp,ws,rp,rs,'s');
[b,a] = butter(n,wn,'s');
[bz,az] = bilinear(b,a,0.5);

z = filter(bz,az,x1);
sound(z,fs);

audiowrite('C:\Users\wangzhiyi\Desktop\test_data2_b.w
av',z,fs);

figure(3);
y3 = fft(z);
```

```
y3 = fftshift(y3);
plot([-fs/2:derta_fs:fs/2-derta_fs],abs(y3));
title('分离波形频域图');
```

六、 结果展示

以下是经过分离后的声音的频域图。其详细的文件在附件里。

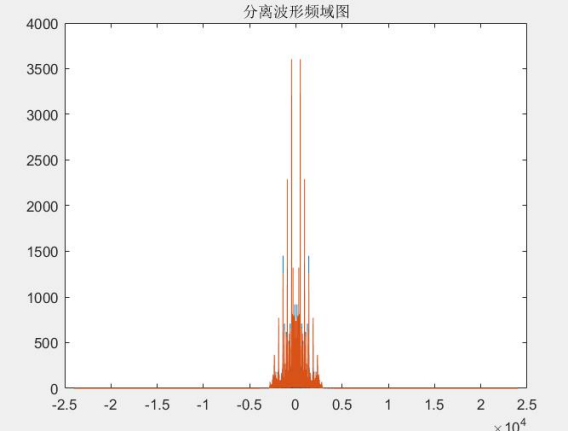


图 24 分离信号频谱图

七、 分析总结

通过听声音，可以听到人的声音可以听见，但是没有了蛐蛐的叫声，但是人的声音失真比较严重。

这道题引用方法为我们在实际生活中对音频信号的分离提出了一种解决办法，例如背景音乐与人声的分离等等。