

# 可视化作业4

18300290007 加兴华

说明：本次作业实现方式均为matlab。附件包含每题的matlab结果生成文件，需将附件中的图片库文件夹移到matlab路径下方可直接运行。

可视化作业4

[1]

(1)

(2)

[2]

## [1]

编程实现基于课件中频率域滤波5步骤的：

(1) 低通平滑操作，并把算法应用与图片上，显示原图的频谱图、频域操作结果的频谱图，以及操作结果；

(2) 实现至少一种图像的锐化操作，该操作是基于频域操作的。

备注：图像的时空-频域变换（即离散傅里叶变换和逆变换）可以调用库函数。

### (1)

第一步：零填充倍增

```
1 function B=fill(A)
2 [m,n]=size(A);
3 B=zeros(2*m,2*n);
4 B(1:m,1:n)=A;
5 end
```

第二步：中心化傅里叶变换

```
1 function B=getf(A)
2 [m,n]=size(A);
3 % 中心平移
4 for i=1:m
5     for j=1:n
6         A(i,j)=(-1)^(i+j)*A(i,j);
7     end
8 end
9 % DFT
10 B=fft2(A);
11 end
```

第三步：生成高斯滤波器

```

1  % 滤波器函数（高斯低通）
2  function B=geth(A,sigma)
3  [m,n]=size(A);
4  B=zeros(2*m,2*n);
5  for x=1:2*m
6      for y=1:2*n
7          r2 = (x-m)^2 + (y-n)^2;
8          B(x,y) = exp(-r2 / (2*sigma^2));
9      end
10 end
11 end
12

```

第四步：点积反变换

```

1  function B=iget(f,h)
2  B=real(ifft2(f.*h));
3  [m,n]=size(B);
4  for x=1:m
5      for y=1:n
6          B(x,y)=B(x,y)*(-1)^(x+y);
7      end
8  end
9  end

```

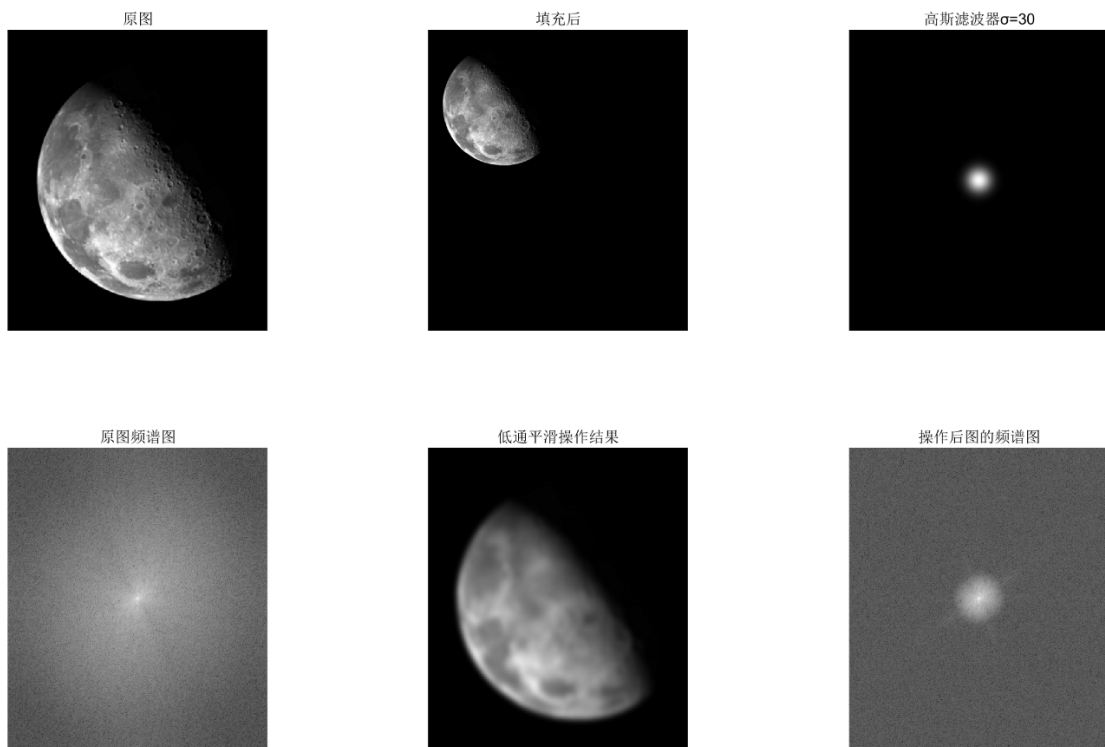
第五步：提取左上限

```

1  function B=cut(A)
2  [m,n]=size(A);
3  B=A(1:m/2,1:n/2);
4  B=uint8(B);
5  end

```

运行附件中基于上述函数而写的lowpass\_fliter.m文件，可以获得如下结果：



## (2)

采用高斯滤波器进行锐化的话，大部分地方与上一问相同，不同点如下：

1，第三步中生成高斯滤波器改为如下

```

1 function B=geth(A,sigma,c)
2 [m,n]=size(A);
3 B=zeros(2*m,2*n);
4 for x=1:2*m
5     for y=1:2*n
6         r2 = (x-m)^2 + (y-n)^2;
7         B(x,y) =1- exp(-r2 / (2*sigma^2));
8     end
9 end
10 end

```

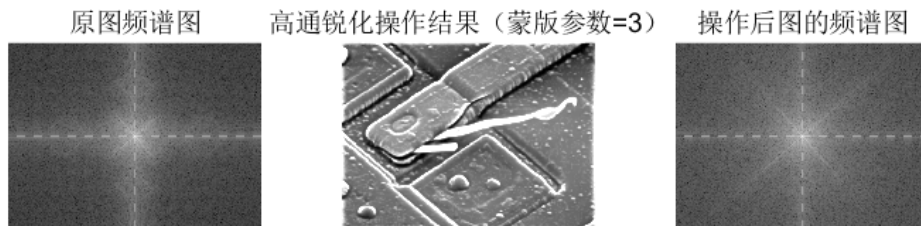
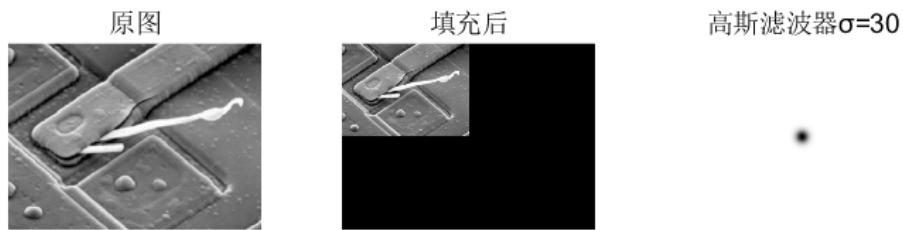
2，通过两次变换得到的结果图在这时仅为蒙版，需要乘以蒙版参数加回原图

```

1 c=cut((iget(f,h)));
2 imshow(A+3*c);

```

运行附件中基于上述函数而写的highpass\_fliter.m文件，可以获得如下结果：



## [2]

编程实现基于频域的选择滤波器方法，去除大脑CT体膜图像（[Shepp-Logan](#)）中的条纹；或自己设计一个有周期噪声的图片，并用频域选择滤波器去除噪声。

备注：图像的时空-频域变换（即离散傅里叶变换和逆变换）可以调用库函数。

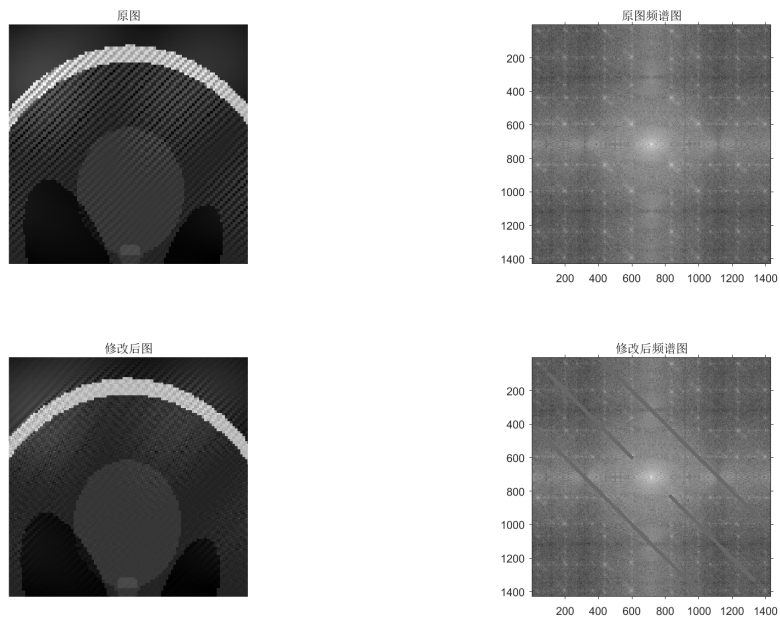
基于前一题代码，稍作修改，即添加cover函数用于遮盖频谱图噪音：

```
1 function B=cover(A,x,y,k)
2 B=A;
3 for i=x-k:x+k
4     for j=y-k:y+k
5         B(i,j)=2000;
6     end
7 end
8 end
```

经过不断调试，我将之应用于与条纹正交的方向：

```
1 for i=100:10:600
2     f=cover(f,i,i,k);
3     f=cover(f,i+400,i,k);
4     f=cover(f,730+i,730+i,k);
5 end
6 for i=20:10:1000
7     f=cover(f,i+400,i,k);
8     f=cover(f,i,i+400,k);
9 end
```

运行附件中基于上述函数而写的removing\_stripes.m文件，可以获得如下结果：



可以观察到条纹在高对比度的地方确实被削弱了，但是原本没有条纹的区域却出现少量条纹，可见单纯遮挡噪音还是有很大的局限性，尤其是当频谱图中的噪音较多时。