**现代图像处理第一次上机**

****

学生姓名： 吴程锴

学 号： 18029100040

班 级： 1802015

授课教师： 王秀美

提交日期：2021年5月12日

目录

[一、 频谱图 1](#_Toc71739091)

[1.1 结果 1](#_Toc71739092)

[1.2 代码 1](#_Toc71739093)

[二、 旋转45度 1](#_Toc71739094)

[2.1 结果 1](#_Toc71739095)

[2.2 代码 2](#_Toc71739096)

[三、 直方图均衡 3](#_Toc71739097)

[3.1 结果 3](#_Toc71739098)

[3.2 代码 3](#_Toc71739099)

[四、 4-邻域平均法 3](#_Toc71739100)

[4.1 结果 3](#_Toc71739101)

[4.2 代码 3](#_Toc71739102)

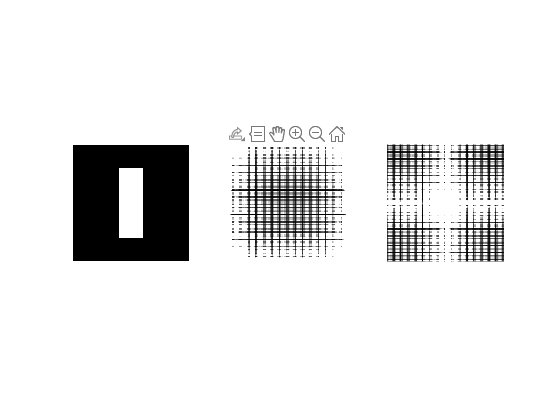
[五、 锐化处理 4](#_Toc71739103)

[5.1 结果 4](#_Toc71739104)

[5.2 代码 4](#_Toc71739105)

# 频谱图

## 结果

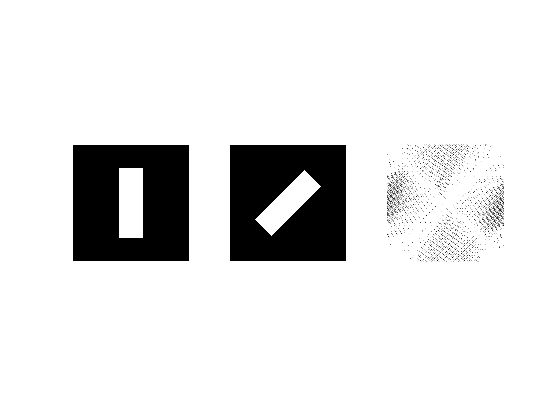


## 代码

1. image1=zeros(500,500);
2. image1(100:400,200:300)=1;
3. figure()
4. subplot(1,3,1)
5. imshow(image1);
6. %% 2D-DFT
7. image2=fft2(image1);
8. subplot(1,3,2)
9. imshow(abs(image2));
10. %% 中心化
11. image3=image1;
12. **for** m=1:size(image3,1)
13. **for** n=1:size(image3,2)
14. image3(m,n)=image3(m,n)\*(-1)^(m+n);
15. end
16. end
17. image3=fft2(image3);
18. subplot(1,3,3)
19. imshow(abs(image3));

# 旋转45度

## 结果



## 代码

1. %% 旋转45°
2. degree=-45;
3. rotate45=[cosd(degree),-sind(degree);sind(degree),cosd(degree)];
4. % image4=image1\*rotate;
5. image4=zeros(size(image1,1),size(image1,2));
6. c=[size(image1,1)/2,size(image1,2)/2];
7. **for** m=1:size(image1,1)
8. **for** n=1:size(image1,2)
9. p=[m;n];
10. pp = round(rotate45\*(p-c)+c);
11. **if** (pp(1) >= 1 && pp(1) <= size(image1,1) && pp(2) >= 1 && pp(2) <= size(image1,2))
12. image4(pp(1),pp(2)) = image1(m, n);
13. end
14. end
15. end
16. %填充
17. **for** m=2:size(image1,1)-1
18. **for** n=2:size(image1,2)-1
19. **if** sum(sum(image4(m-1:m+1,n-1:n+1)))>=4
20. image4(m,n)=1;
21. end
22. end
23. end
24. figure()
25. subplot(1,3,1)
26. imshow(image1)
27. subplot(1,3,2)
28. imshow(image4)
29. %%
30. image5=fft2(image4);
31. subplot(1,3,3)
32. imshow(abs(image5));

# 直方图均衡

## 结果



## 代码

1. %% lena
2. lena=imread('lena.png');
3. lena=**double**(rgb2gray(lena))/255;
4. %% 直方图均衡
5. lena\_balancing=histeq(lena);
6. figure()
7. subplot(1,2,1)
8. imshow(lena);
9. subplot(1,2,2)
10. imshow(lena\_balancing);

# 4-邻域平均法

## 结果



## 代码

1. %% 加噪声
2. lena\_noise=lena+0.1\*randn(size(lena,1),size(lena,2));
3. figure()
4. subplot(1,3,1)
5. imshow(lena);
6. subplot(1,3,2)
7. imshow(lena\_noise)
8. %% 4-邻域平均法
9. lena\_filter=lena\_noise;
10. **for** m=2:size(lena\_filter,1)-1
11. **for** n=2:size(lena\_filter,2)-1
12. lena\_filter(m,n)=(lena\_noise(m-1,n)+lena\_noise(m+1,n)+lena\_noise(m,n-1)+lena\_noise(m,n+1))/4;
13. end
14. end
15. subplot(1,3,3)
16. imshow(lena\_filter)

# 锐化处理

## 结果



## 代码

1. %% 锐化
2. a=2;
3. W=[0,-a,0;-a,1+4\*a,-a;0,-a,0];
4. lena\_sharpen=lena;
5. **for** m=2:size(lena\_sharpen,1)-1
6. **for** n=2:size(lena\_sharpen,2)-1
7. lena\_sharpen(m,n)=sum(sum(lena(m-1:m+1,n-1:n+1).\*W));
8. end
9. end
10. figure()
11. subplot(1,2,1)
12. imshow(lena);
13. subplot(1,2,2)
14. imshow(lena\_sharpen);