**姓名：郑天骥 学号：01170227 专业：计算机应用技术**

**1 编程实现直方图均衡**

clc;

image = imread('pout.tif');

[height, width] = size(image);

NumPixel = zeros(1,256); % 建立一个256列的行向量，以统计各灰度级的像素个数

for i = 1 : height

for j = 1 : width

k = image(i,j); % k是像素点(i,j)的灰度值

NumPixel(k+1) = NumPixel(k+1) + 1; % 对应灰度值像素点数量加1

end

end

ProbPixel = zeros(1,256); % 统计各灰度级出现的频率

for i = 1 : 256

ProbPixel(i) = NumPixel(i) / (height \* width);

end

CumPixel = cumsum(ProbPixel); % 这里的数组CumPixel大小也是1×256

CumPixel = uint8((256-1) .\* CumPixel + 0.5);

outImage = uint8(zeros(height, width)); % 预分配数组

for i = 1 : height

for j = 1 : width

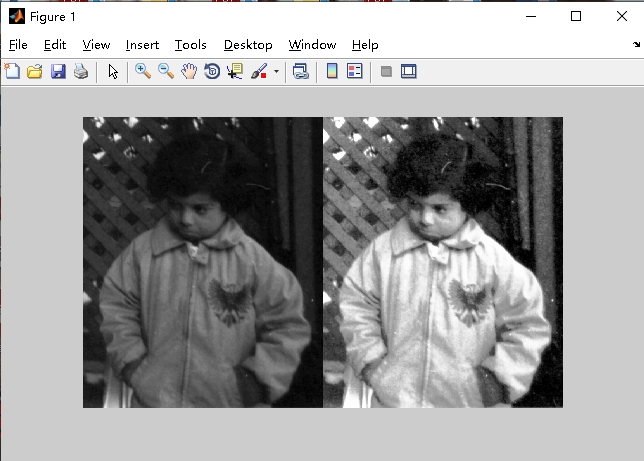
outImage(i,j) = CumPixel(image(i,j));

end

end

% 显示直方图均衡化前后的图像

imshowpair(image, outImage, 'montage');



**2图像平滑的实现，不限定滤波器**

clc;

I = imread('coins.png');

Inoised = imnoise(I,'gaussian',0.1,0.005);%对图像进行高斯噪声加噪

%制定卷积核

h=ones(3,3)/5;

h(1,1) = 0;

h(1,3) = 0;

h(3,1) = 0;

h(1,3) = 0;

%平滑运算

I2=imfilter(Inoised,h);

subplot(1,3,1);imshow(I);

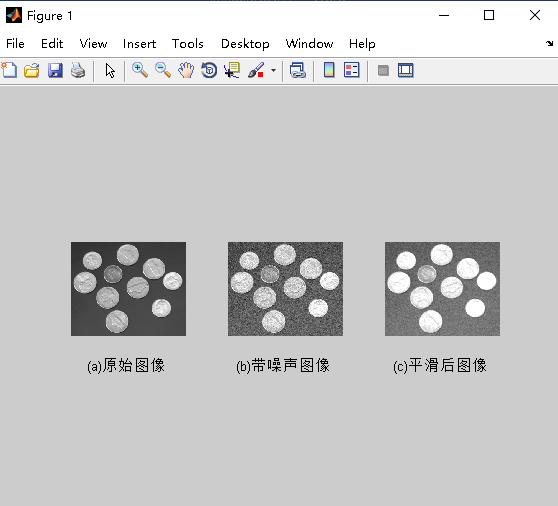
xlabel('(a)原始图像');

subplot(1,3,2);imshow(Inoised);

xlabel('(b)带噪声图像');

subplot(1,3,3);imshow(I2);

xlabel('(c)平滑后图像');



**3图像锐化的实现，不限定滤波器**

clc;

I = imread('pout.tif');

J = histeq(I,256);

imshowpair(I,J,'montage');

clear all;

I = imread('cameraman.tif');

subplot(2,2,1);imshow(I);

xlabel('(a)原始图像');

H = fspecial('motion',20,45);

MotionBlur = imfilter(I,H,'replicate');

subplot(2,2,2);imshow(MotionBlur);

xlabel('(b)运动模糊图像');

H=fspecial('disk',10);

blurred = imfilter(I,H,'replicate');

subplot(2,2,3);imshow(blurred);

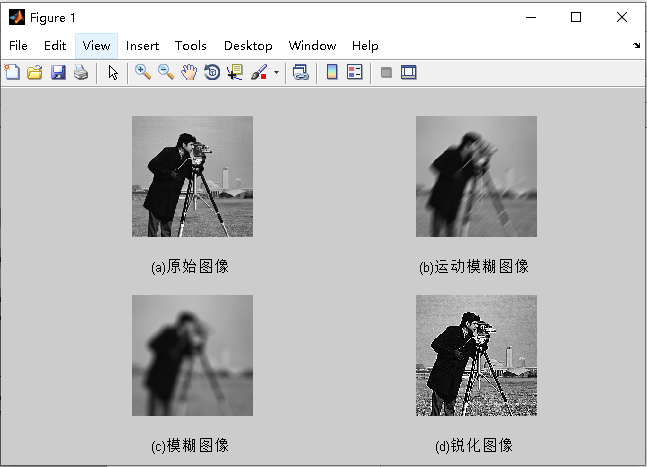
xlabel('(c)模糊图像');

H=fspecial('unsharp');

sharpened = imfilter(I,H,'replicate');

subplot(2,2,4);imshow(sharpened);

xlabel('(d)锐化图像');



**4频率域图像平滑实现， 不限定滤波器**

f0=imread('pout.tif');

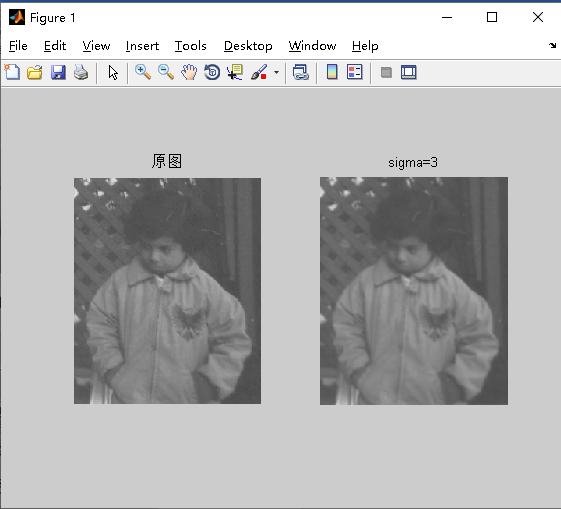
subplot(1,2,1);imshow(f0);title('原图');

sigma=3;

gausFilter=fspecial('gaussian',[3,3],sigma);

gaus=imfilter(f0,gausFilter,'replicate');

subplot(1,2,2);imshow(gaus);title('sigma=3');



**5频率域图像锐化实现，不限定滤波器**

f = imread('tire.tif')

f=double(f);

[m,n]=size(f);

%a

figure(1)

imshow(f,[]);

title('origin')

p=2\*m;

q=2\*n;

A=zeros(p,q);

for i=1:m

for j=1:n

A(i,j)=f(i,j);

end

end

for i=1:p

for j=1:q

A(i,j)=A(i,j)\*(-1)^(i+j);

end

end

I2=fft2(A);

for i=1:p

for j=1:q

D(i,j)=sqrt((i-p/2)^2+(j-q/2)^2);

end

end

X=zeros(p,q);

%pi=3.14

for i=1:p

for j=1:q

X(i,j)=-4\*pi^2\*D(i,j)^2;

Y(i,j)=X(i,j)\*I2(i,j);

end

end

gp3=real(ifft2(Y));

for i=1:p

for j=1:q

gp3(i,j)=gp3(i,j).\*(-1).^(i+j);

end

end

figure(3);

imshow(gp3,[]);

gp3max = max(gp3(:));

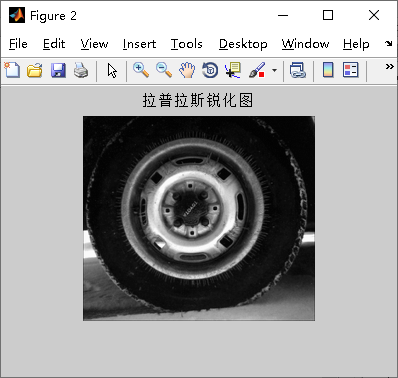
g3 = gp3(1:m,1:n);

k=10;

gs = f-k\*g3/gp3max;

figure(2)

imshow(gs,[]),title('拉普拉斯锐化图')



**6 RGB锐化**

clc;

rgb1= imread('football.jpg');

rgb=im2double(rgb1);

rgb\_R=rgb(:,:,1);

rgb\_G= rgb(:,:,2);

rgb\_B= rgb(:,:,3);

lapMatrix=[1 1 1;1 -8 1;1 1 1];%模板1 1\*8

Matrix2=[1 0 1;0 -4 0;1 0 1];%模板21\*4

%模板1\*8 3个不同的第三参数的锐化滤波

f\_R=imfilter(rgb\_R,lapMatrix,'replicate');

f\_G=imfilter(rgb\_G,lapMatrix,'replicate');

f\_B=imfilter(rgb\_B,lapMatrix,'replicate');

f\_R2=imfilter(rgb\_R,lapMatrix,'symmetric');

f\_G2=imfilter(rgb\_G,lapMatrix,'symmetric');

f\_B2=imfilter(rgb\_B,lapMatrix,'symmetric');

f\_R3=imfilter(rgb\_R,lapMatrix,'circular');

f\_G3=imfilter(rgb\_G,lapMatrix,'circular');

f\_B3=imfilter(rgb\_B,lapMatrix,'circular');

%模板 1\*4 3个不同的第三参数的锐化滤波

f2\_R=imfilter(rgb\_R,Matrix2,'replicate');

f2\_G=imfilter(rgb\_G,Matrix2,'replicate');

f2\_B=imfilter(rgb\_B,Matrix2,'replicate');

f2\_R2=imfilter(rgb\_R,Matrix2,'symmetric');

f2\_G2=imfilter(rgb\_G,Matrix2,'symmetric');

f2\_B2=imfilter(rgb\_B,Matrix2,'symmetric');

f2\_R3=imfilter(rgb\_R,Matrix2,'circular');

f2\_G3=imfilter(rgb\_G,Matrix2,'circular');

f2\_B3=imfilter(rgb\_B,Matrix2,'circular');

%模板1 锐化后的3分量

rgb\_tmp=cat(3,f\_R,f\_G,f\_B);

rgb\_tmp2=cat(3,f\_R2,f\_G2,f\_B2);

rgb\_tmp3=cat(3,f\_R3,f\_G3,f\_B3);

%模板2 锐化后的3分量

rgb2\_tmp=cat(3,f2\_R,f2\_G,f2\_B);

rgb2\_tmp2=cat(3,f2\_R2,f2\_G2,f2\_B2);

rgb2\_tmp3=cat(3,f2\_R3,f2\_G3,f2\_B3);

%模板1 3种参数的处理结果与原图相减

rgb\_sharped =imsubtract(rgb,rgb\_tmp);

rgb\_sharped2=imsubtract(rgb,rgb\_tmp2);

rgb\_sharped3=imsubtract(rgb,rgb\_tmp3);

%模板2 3种参数的处理结果与原图相减

rgb2\_sharped =imsubtract(rgb,rgb2\_tmp);

rgb2\_sharped2=imsubtract(rgb,rgb2\_tmp2);

rgb2\_sharped3=imsubtract(rgb,rgb2\_tmp3);

figure;

subplot(341);imshow(rgb);title('原图');

subplot(342);imshow(rgb\_R);title('R');

subplot(343);imshow(rgb\_G);title('G');

subplot(344);imshow(rgb\_B);title('B');

subplot(345);imshow(rgb\_tmp);title('cat\_r\_g\_b ');

subplot(346);imshow(rgb\_tmp2);title('cat\_r\_g\_b2');

subplot(347);imshow(rgb\_tmp3);title('cat\_r\_g\_b3');

subplot(348);imshow(rgb\_sharped);title('1\*8 replicate');

subplot(349);imshow(rgb\_sharped2);title('1\*8 symmetric');

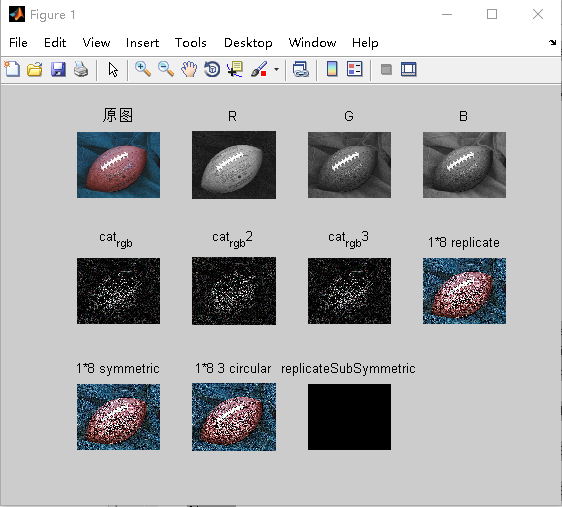
subplot(3,4,10);imshow(rgb\_sharped3);title('1\*8 3 circular');

rgb\_diff=imsubtract(rgb\_sharped,rgb\_sharped2);%replicate sub symmetric

subplot(3,4,11);imshow(rgb\_diff);title('replicateSubSymmetric');

figure;

imshow(rgb3\_diff);title('1\*8Sub1\*4');



**7 HIS锐化**

function rgb = hsi2rgb(hsi)

H = hsi(:, :, 1) \* 2 \* pi;

S = hsi(:, :, 2);

I = hsi(:, :, 3);

R = zeros(size(hsi, 1), size(hsi, 2));

G = zeros(size(hsi, 1), size(hsi, 2));

B = zeros(size(hsi, 1), size(hsi, 2));

idx = find( (0 <= H) & (H < 2\*pi/3));

B(idx) = I(idx) .\* (1 - S(idx));

R(idx) = I(idx) .\* (1 + S(idx) .\* cos(H(idx)) ./ cos(pi/3 - H(idx)));

G(idx) = 3\*I(idx) - (R(idx) + B(idx));

idx = find( (2\*pi/3 <= H) & (H < 4\*pi/3) );

R(idx) = I(idx) .\* (1 - S(idx));

G(idx) = I(idx) .\* (1 + S(idx) .\* cos(H(idx) - 2\*pi/3) ./ cos(pi - H(idx)));

B(idx) = 3\*I(idx) - (R(idx) + G(idx));

idx = find( (4\*pi/3 <= H) & (H <= 2\*pi));

G(idx) = I(idx) .\* (1 - S(idx));

B(idx) = I(idx) .\* (1 + S(idx) .\* cos(H(idx) - 4\*pi/3) ./cos(5\*pi/3 - H(idx)));

R(idx) = 3\*I(idx) - (G(idx) + B(idx));

rgb = cat(3, R, G, B);

rgb = max(min(rgb, 1), 0);

rgb1= imread('football.jpg');

rgb=im2double(rgb1);

rgb\_R=rgb(:,:,1);

rgb\_G= rgb(:,:,2);

rgb\_B= rgb(:,:,3);

lapMatrix=[1 1 1;1 -8 1;1 1 1];

Matrix2=[1 0 1;0 -4 0;1 0 1];

%模板21\*4

%模板1\*8 3个不同的第三参数的锐化滤波

f\_R=imfilter(rgb\_R,lapMatrix,'replicate');

f\_G=imfilter(rgb\_G,lapMatrix,'replicate');

f\_B=imfilter(rgb\_B,lapMatrix,'replicate');

%symmetric

f\_R2=imfilter(rgb\_R,lapMatrix,'symmetric');

f\_G2=imfilter(rgb\_G,lapMatrix,'symmetric');

f\_B2=imfilter(rgb\_B,lapMatrix,'symmetric');

%circular

f\_R3=imfilter(rgb\_R,lapMatrix,'circular');

f\_G3=imfilter(rgb\_G,lapMatrix,'circular');

f\_B3=imfilter(rgb\_B,lapMatrix,'circular');

%模板 1\*4 3个不同的第三参数的锐化滤波

f2\_R=imfilter(rgb\_R,Matrix2,'replicate');

f2\_G=imfilter(rgb\_G,Matrix2,'replicate');

f2\_B=imfilter(rgb\_B,Matrix2,'replicate');

f2\_R2=imfilter(rgb\_R,Matrix2,'symmetric');

f2\_G2=imfilter(rgb\_G,Matrix2,'symmetric');

f2\_B2=imfilter(rgb\_B,Matrix2,'symmetric');

f2\_R3=imfilter(rgb\_R,Matrix2,'circular');

f2\_G3=imfilter(rgb\_G,Matrix2,'circular');

f2\_B3=imfilter(rgb\_B,Matrix2,'circular');

%模板1 锐化后的3分量

rgb\_tmp=cat(3,f\_R,f\_G,f\_B);

rgb\_tmp2=cat(3,f\_R2,f\_G2,f\_B2);

rgb\_tmp3=cat(3,f\_R3,f\_G3,f\_B3);

%模板2 锐化后的3分量

rgb2\_tmp=cat(3,f2\_R,f2\_G,f2\_B);

rgb2\_tmp2=cat(3,f2\_R2,f2\_G2,f2\_B2);

rgb2\_tmp3=cat(3,f2\_R3,f2\_G3,f2\_B3);

%模板1 3种参数的处理结果与原图相减

rgb\_sharped =imsubtract(rgb,rgb\_tmp);

rgb\_sharped2=imsubtract(rgb,rgb\_tmp2);

rgb\_sharped3=imsubtract(rgb,rgb\_tmp3);

%模板2 3种参数的处理结果与原图相减

rgb2\_sharped =imsubtract(rgb,rgb2\_tmp);

rgb2\_sharped2=imsubtract(rgb,rgb2\_tmp2);

rgb2\_sharped3=imsubtract(rgb,rgb2\_tmp3);

figure;

subplot(341);imshow(rgb);title('原图');

subplot(342);imshow(rgb\_R);title('R');

subplot(343);imshow(rgb\_G);title('G');

subplot(344);imshow(rgb\_B);title('B');

subplot(345);imshow(rgb2\_tmp);title('cat\_r\_g\_b ');

subplot(346);imshow(rgb2\_tmp2);title('cat\_r\_g\_b2');

subplot(347);imshow(rgb2\_tmp3);title('cat\_r\_g\_b3');

subplot(348);imshow(rgb2\_sharped);title('1\*4 replicate');

subplot(349);imshow(rgb2\_sharped2);title('1\*4 symmetric');

subplot(3,4,10);imshow(rgb2\_sharped3);title('1\*4 3 circular');

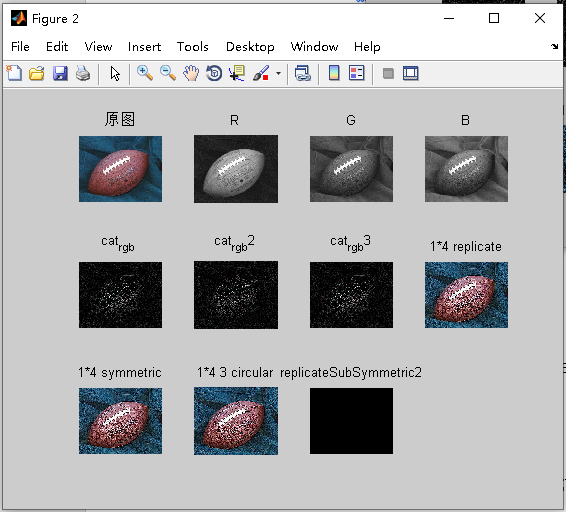
rgb2\_diff=imsubtract(rgb2\_sharped,rgb2\_sharped2);%replicate sub symmetric

subplot(3,4,11);imshow(rgb2\_diff);title('replicateSubSymmetric2');

rgb3\_diff=imsubtract(rgb2\_sharped2,rgb\_sharped2);%1\*8Sub1\*4

figure;

imshow(rgb3\_diff);title('1\*8Sub1\*4');



**8 RGB反色**

clc;

a=imread('woman.jpg');

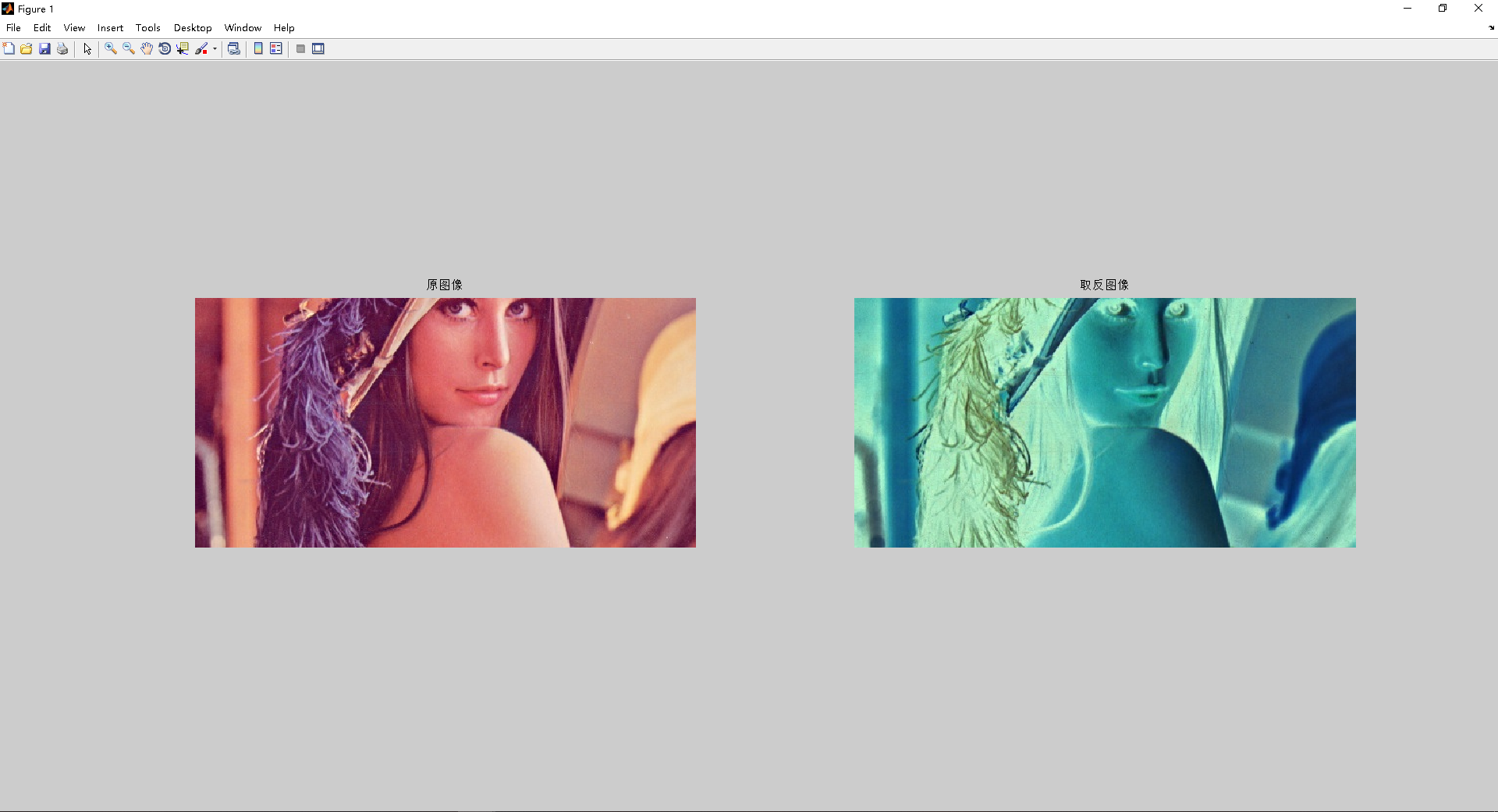
figure(1),subplot(1,2,1);imshow(a);title('原图像');

gary(:,:,1)=255-a(:,:,1);

gary(:,:,2)=255-a(:,:,2);

gary(:,:,3)=255-a(:,:,3);

figure(1),subplot(1,2,2);imshow(gary);title('取反图像');



**9 HIS反色**

function rgb = hsi2rgb(hsi)

H = hsi(:, :, 1) \* 2 \* pi;

S = hsi(:, :, 2);

I = hsi(:, :, 3);

R = zeros(size(hsi, 1), size(hsi, 2));

G = zeros(size(hsi, 1), size(hsi, 2));

B = zeros(size(hsi, 1), size(hsi, 2));

idx = find( (0 <= H) & (H < 2\*pi/3));

B(idx) = I(idx) .\* (1 - S(idx));

R(idx) = I(idx) .\* (1 + S(idx) .\* cos(H(idx)) ./ cos(pi/3 - H(idx)));

G(idx) = 3\*I(idx) - (R(idx) + B(idx));

idx = find( (2\*pi/3 <= H) & (H < 4\*pi/3) );

R(idx) = I(idx) .\* (1 - S(idx));

G(idx) = I(idx) .\* (1 + S(idx) .\* cos(H(idx) - 2\*pi/3) ./ cos(pi - H(idx)));

B(idx) = 3\*I(idx) - (R(idx) + G(idx));

idx = find( (4\*pi/3 <= H) & (H <= 2\*pi));

G(idx) = I(idx) .\* (1 - S(idx));

B(idx) = I(idx) .\* (1 + S(idx) .\* cos(H(idx) - 4\*pi/3) ./cos(5\*pi/3 - H(idx)));

R(idx) = 3\*I(idx) - (G(idx) + B(idx));

rgb = cat(3, R, G, B);

rgb = max(min(rgb, 1), 0);

a=imread('woman.jpg');

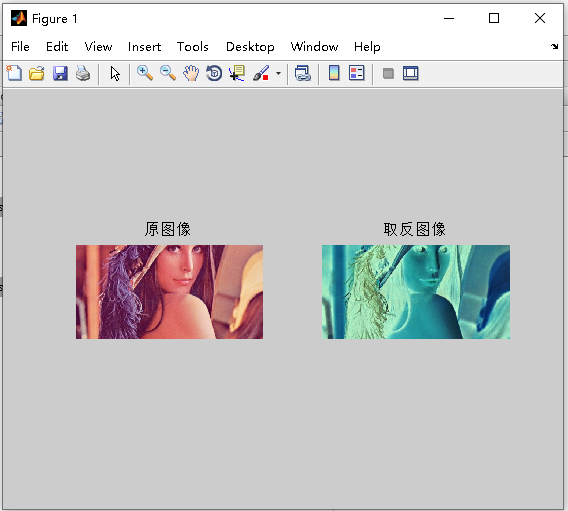
figure(1),subplot(1,2,1);imshow(a);title('原图像');

gary(:,:,1)=255-a(:,:,1);

gary(:,:,2)=255-a(:,:,2);

gary(:,:,3)=255-a(:,:,3);

figure(1),subplot(1,2,2);imshow(gary);title('取反图像');



**10形态学开运算的编程实现**

I1=imread('tire.tif'); %读灰度图tire.tif

I2=imnoise(I1,'salt & pepper'); %在图像上加入椒盐噪声

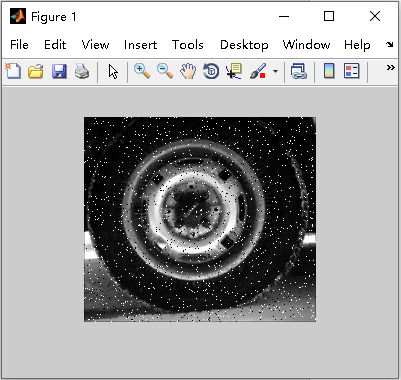
figure,imshow(I2) %显示加椒盐噪声后的灰度图像

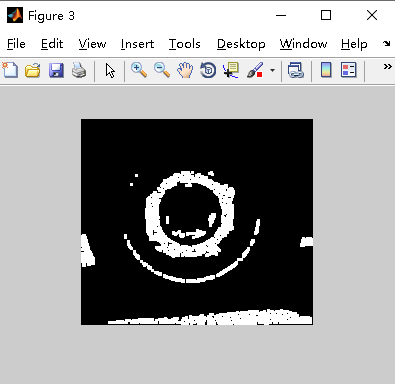
I3=im2bw(I2);

figure,imshow(I3) %显示二值化后的图像

I4=bwmorph(I3,'open'); %对二值噪声图像进行二值形态学开运算

figure,imshow(I4) %显示开运算后的图像





**11用Hough变换 检测直线**

clc;

r = imread('line1.jpg');

t=rgb2gray(r);

figure,imshow(t)

f=edge(t,'canny');

figure,imshow(f)

[H, theta, rho] = hough(f,'RhoResolution',0.5,'Theta',-90:0.5:89.5);

figure

imshow(imadjust(H),'Xdata', theta, 'Ydata', rho, 'InitialMagnification','fit');

axis on, axis normal

p= houghpeaks(H, 5);

%[r,c] = houghpeaks(H, 5);

hold on;

%plot(theta(r), rho(c), 'linestyle', 'none', 'marker', 's', 'color', 'w')

plot(p, 'linestyle', 'none', 'marker', 's', 'color', 'r')

lines = houghlines(f, theta, rho,p);

%lines = houghlines(f, theta, rho, r, c);

figure,

imshow(t)

hold on

for k = 1: length(lines)

xy = [lines(k).point1; lines(k).point2];

plot(xy(:, 2), xy(:, 1), 'LineWidth', 4, 'Color', [.6, .2, .1]);

end

hold off

