数字图像处理第七次作业报告

吴志朋
自动化 62
2160504050

摘要：数字图像处理是通过计算机对图像进行去除噪声、增强、复原、分割、提取特征等处理的方法，然后利用计算机达到某种目的的技术。数字图像处理技术是在计算机发展和数学发展的推动下，得以进步和完善的，最初的时候主要应用在生物医药工程和工业工程等方面。其实图像处理技术一项具有良好发展前景的新型技术，在未来的不断进步发展过程中，必然会对社会做出巨大贡献。本文实现了图像的空间滤波和边缘检测。

一、边缘检测

边缘是所要提取目标和背景的边界线，而提取出边缘才能将目标和背景区分出来。因此，边缘检测技术对数字图像处理非常重要。常用的边缘检测算子有 sobel，Roberts，Prewitt,Laplace 等等

1)Robert 算子

Robert 使用一个 2×2 的模板，对角线做差，其差分为：

$$g_x = (z_9 - z_5) \text{ 和 } g_y = (z_8 - z_6)$$

因为向量无法在图像中显示，我们要计算梯度向量的长度：

$$M(x,y) = \left[(z_9 - z_5)^2 + (z_8 - z_6)^2\right]^{1/2}$$

简化为绝对值方法：$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|$$

这个就是 Robert 交叉算子。模板：

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

2）Sobel 模板：

其数字模板为：

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

其下降速率（梯度的长度）计算公式：

$$M(x,y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)| + |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

3）canny 算子：

Canny 算子求边缘点具体算法步骤如下：

用高斯滤波器平滑图像.

用一阶偏导有限差分计算梯度幅值和方向.

对梯度幅值进行非极大值抑制.

用双阈值算法检测和连接边缘.

二、直线检测

hough 变换原理：

x-y 坐标系中直线方程为 y=ax+b x-y 坐标系中的每一个点在，在 a-b 坐标系中对应一条直线 x-y 坐标系中共线的点集，在 a-b 坐标系中对应一个线束（射影几何中的概念），线束的中心(a′,b′)，在 x-y 平面中可以确定一条直线

但是直线方程 y=ax+b 不能表示斜率无限大的直线，所用改用直线方程：

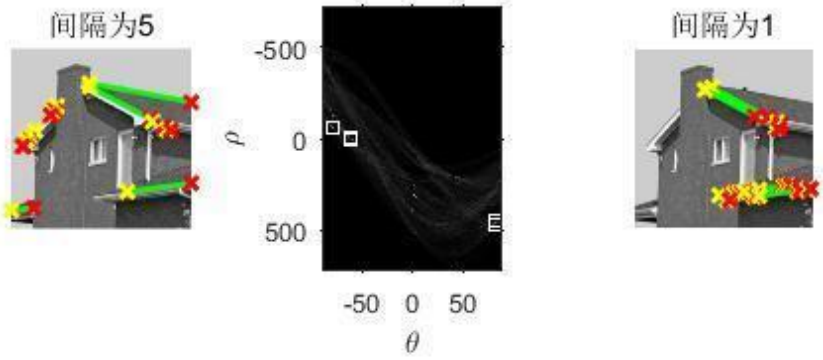
xcosθ+ysinθ=ρ x-y 坐标系上的每一点在参数空间 θ-ρ 对应一条正弦线 x-y 坐标系上共线的点集，在 θ-ρ 空间中有一组正弦线，并且交于一点(θ′,ρ′)，在 x-y 平面中可确定一条直线

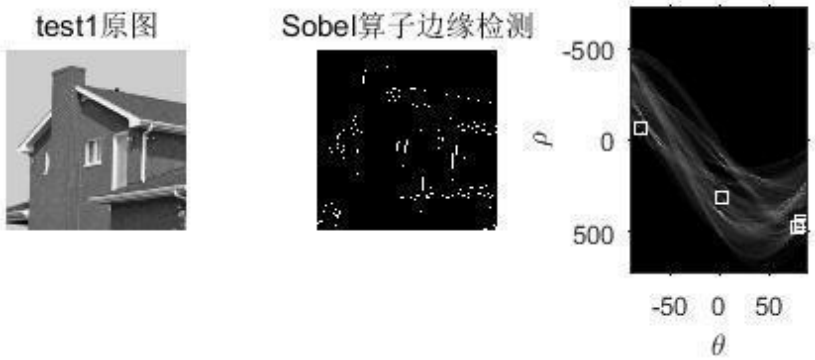
将图像空间（x-y 平面）中所有点映射到 θ-ρ 空间（实际上是二维矩阵），这个二维矩阵叫做“累计数组”或“累加器”。累计数组中值较大的元素，往往对应着图像中的直线。

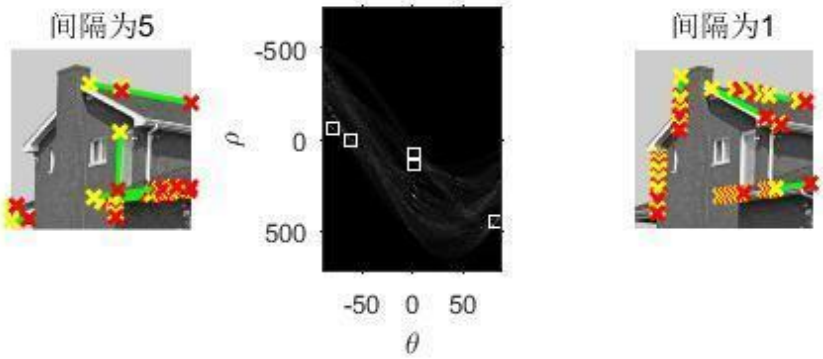
三 比较不同边缘检测算法（2 种以上）、不同 hough 变换参数对直线检测的影响；

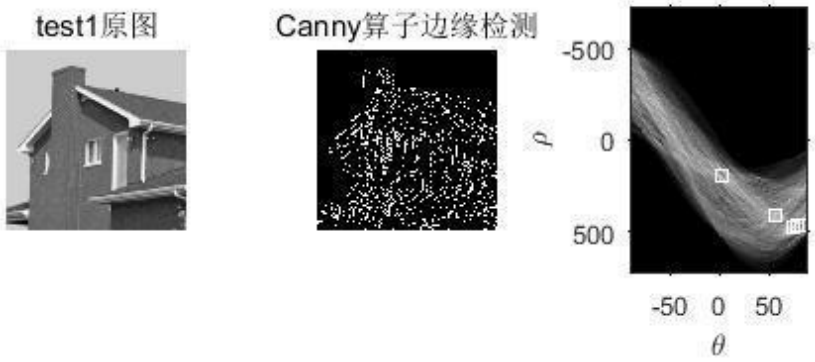
test1：

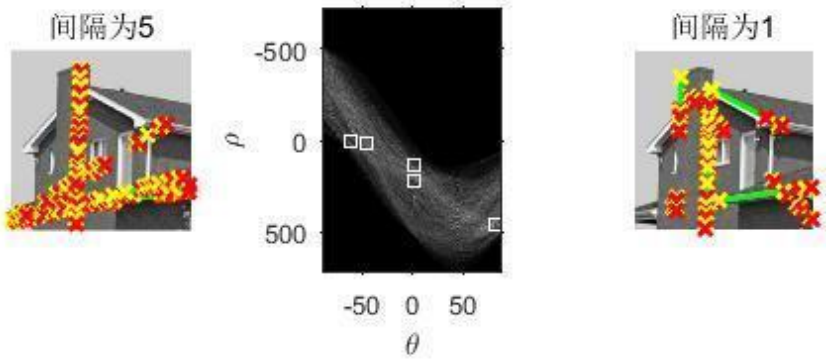

test1原图
Robert算子边缘检测
-500
0
500
ρ
-50 0 50
θ


间隔为5
-500
0
500
ρ
-50 0 50
θ
间隔为1

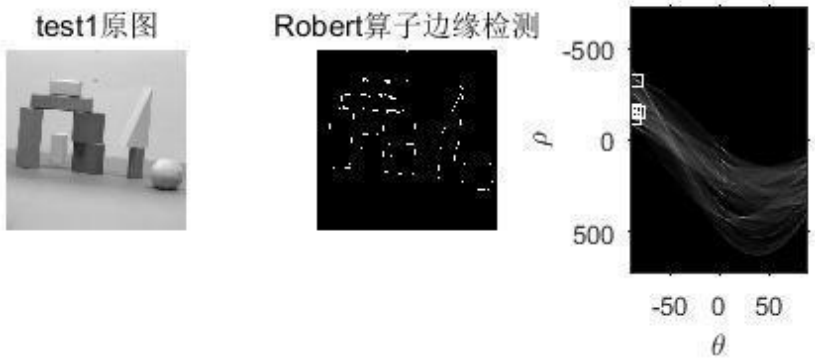

test1原图
Sobel算子边缘检测
-500
0
500
ρ
-50 0 50
θ

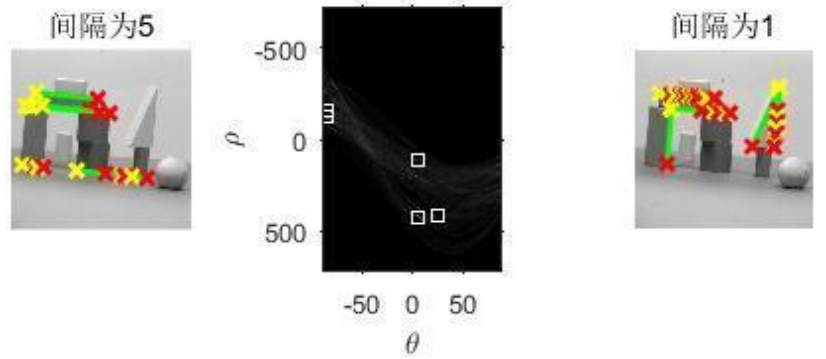

间隔为5
-500
0
500
ρ
-50 0 50
θ
间隔为1


test1原图
Canny算子边缘检测
-500
0
500
ρ
-50 0 50
θ


间隔为5
-500
0
500
ρ
-50 0 50
θ
间隔为1

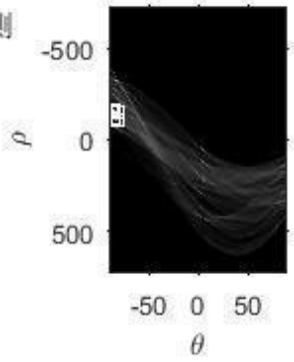
test2：


test1原图
Robert算子边缘检测
-500
0
500
ρ
-50 0 50
θ

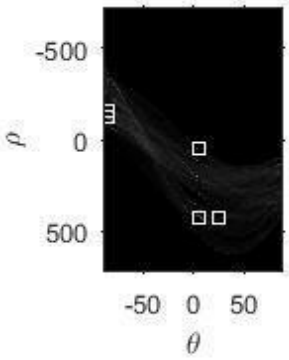

间隔为5
-500
0
500
ρ
-50 0 50
θ
间隔为1

test1原图

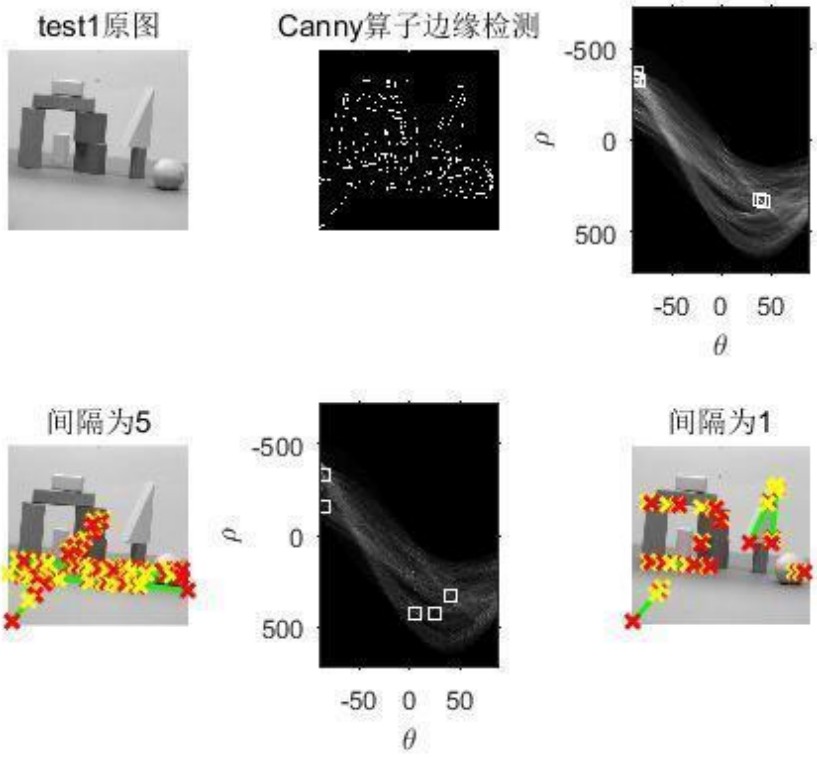
Sobel算子边缘检测



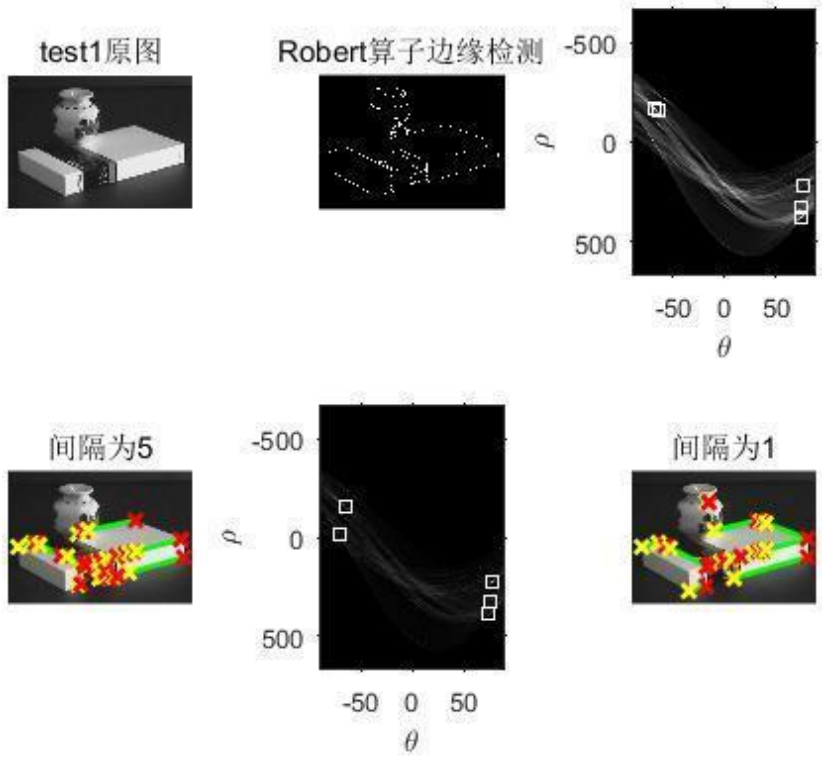
间隔为5

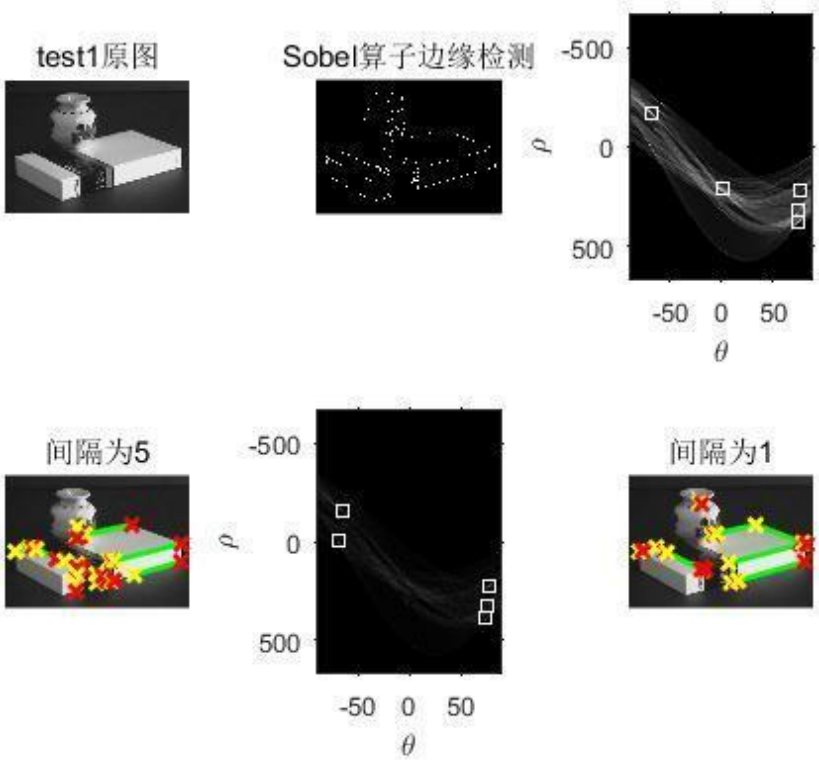


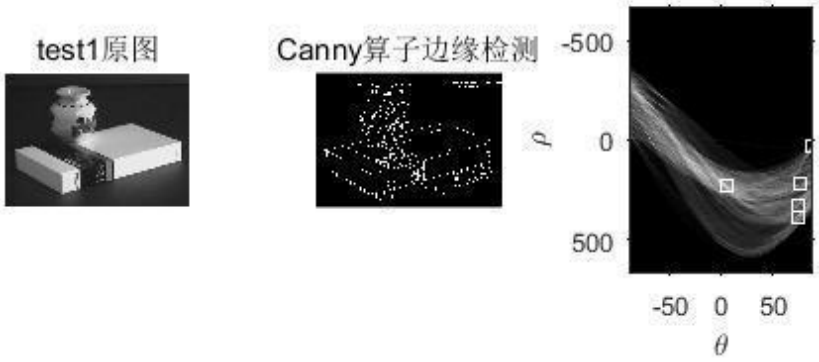
间隔为1

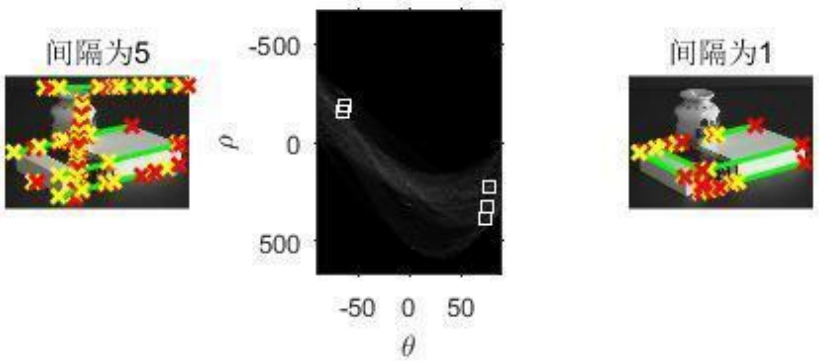

test1原图
Canny算子边缘检测
-500
0
500
ρ
-50 0 50
θ
间隔为5
间隔为1

test3：


test1原图
Robert算子边缘检测
-500
0
500
ρ
-50 0 50
θ
间隔为5
间隔为1

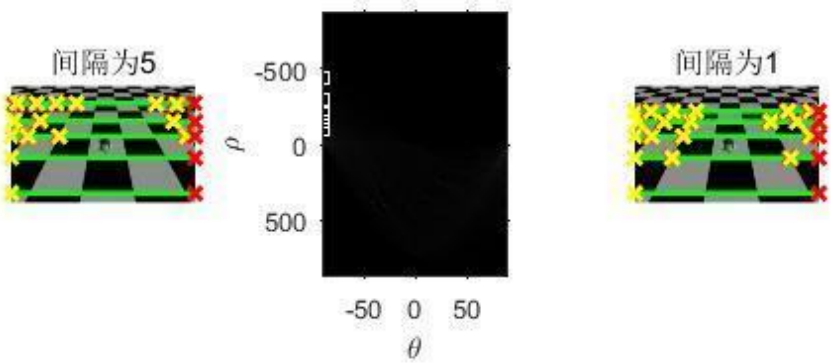

test1原图
Sobel算子边缘检测
-500
0
500
ρ
-50
0
50
θ
间隔为5
-500
0
500
ρ
-50
0
50
θ
间隔为1

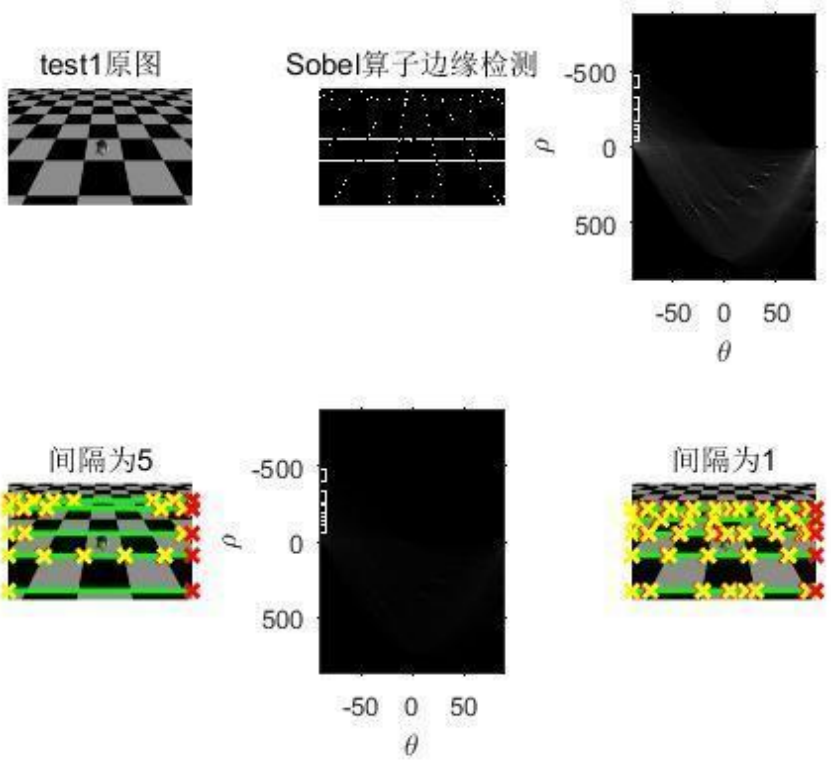

test1原图
Canny算子边缘检测
-500
0
500
ρ
-50
0
50
θ

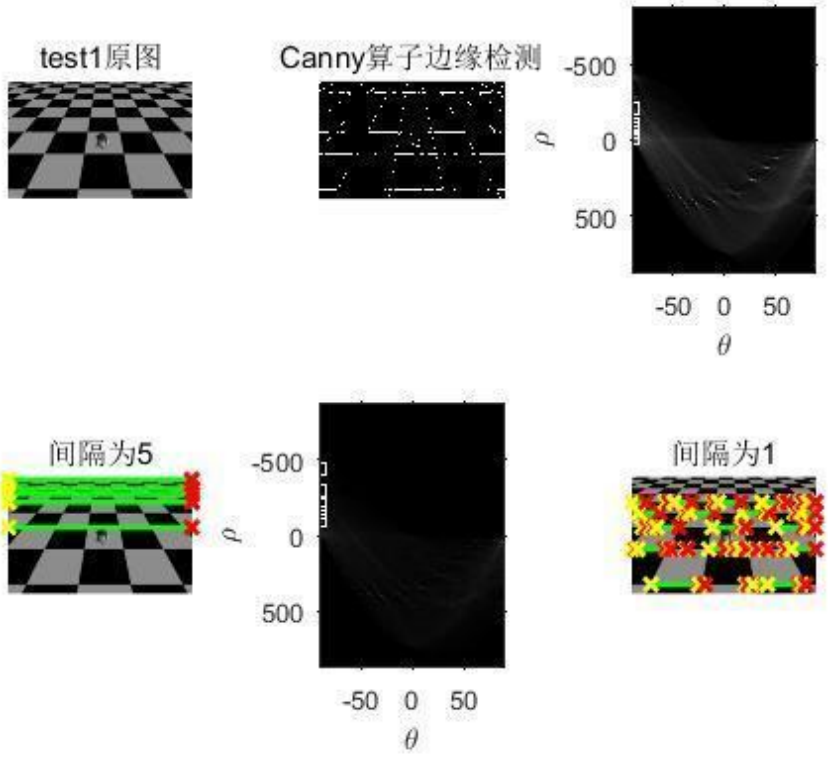

间隔为5
-500
0
500
ρ
-50
0
50
θ
间隔为1

test4：

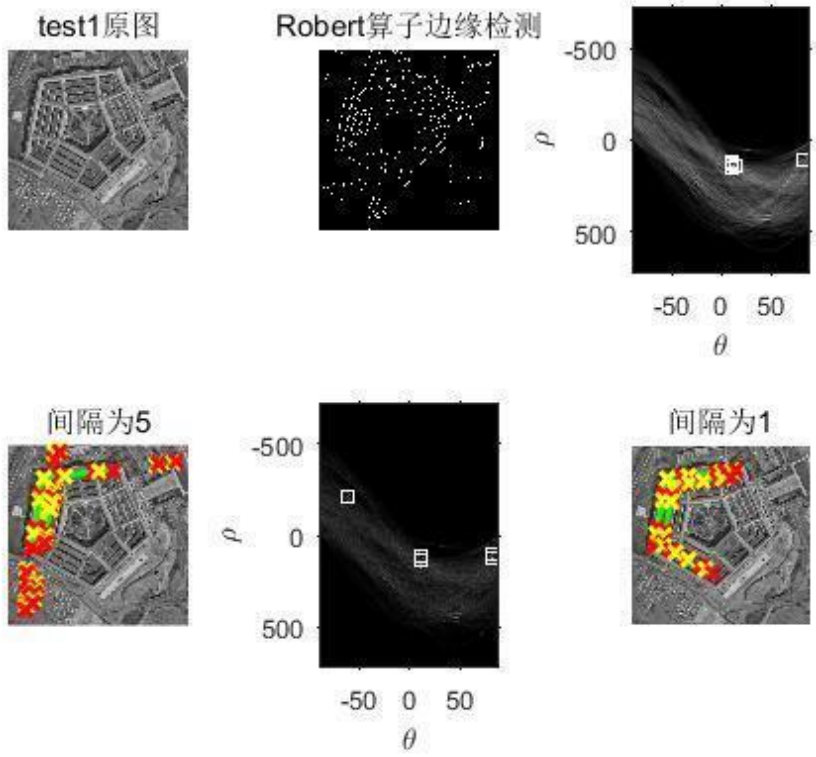

test1原图
Robert算子边缘检测
-500
0
500
ρ
-50
0
50
θ

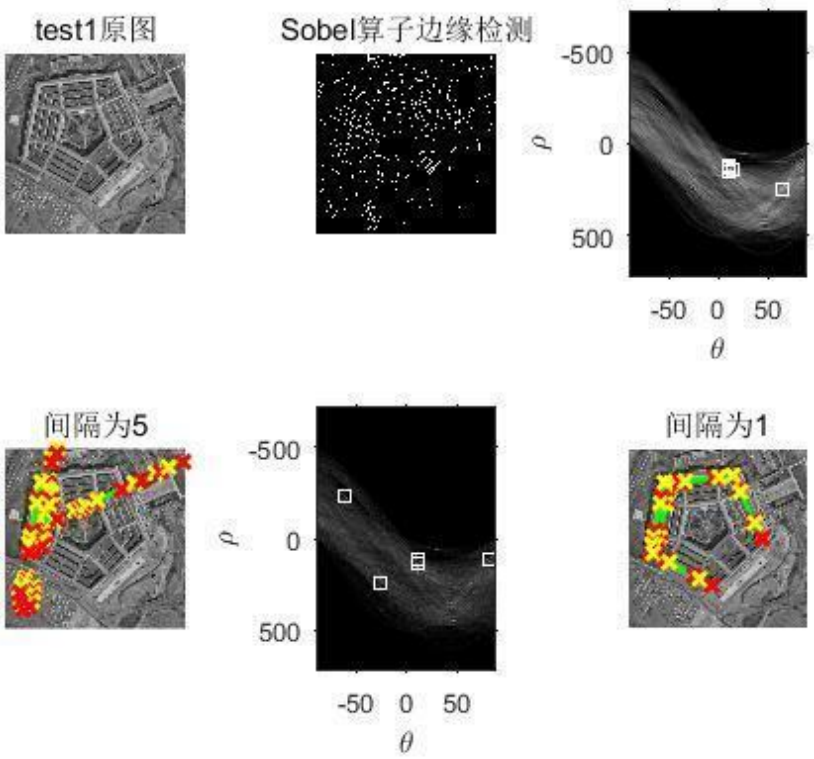

间隔为5
-500
0
500
ρ
-50
0
50
θ
间隔为1

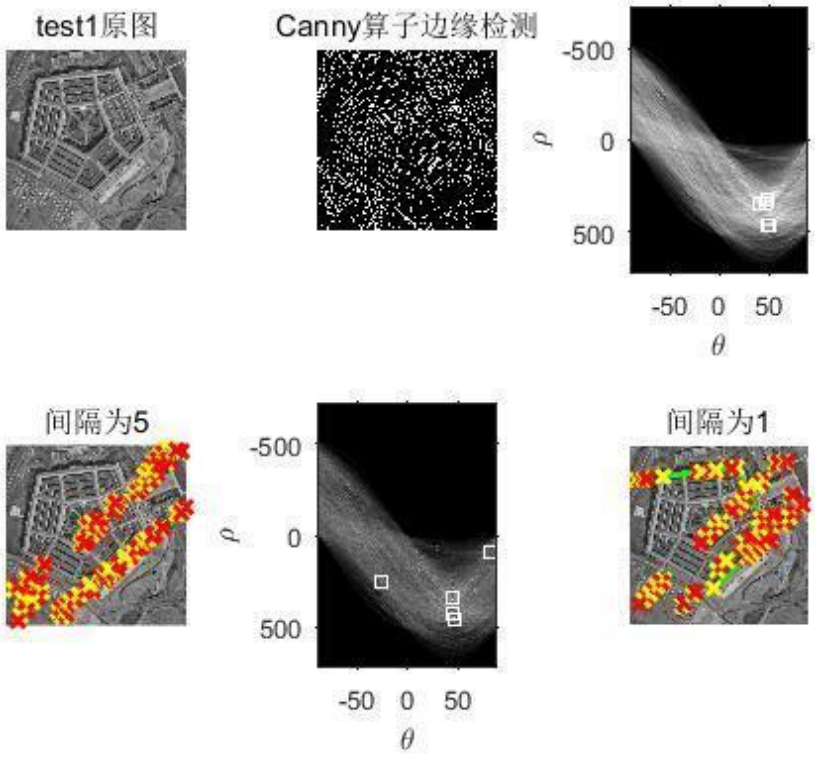

test1原图
Sobel算子边缘检测
-500
0
500
ρ
-50
0
50
θ
间隔为5
间隔为1


test1原图
Canny算子边缘检测
-500
0
500
ρ
-50 0 50
θ
间隔为5
间隔为1

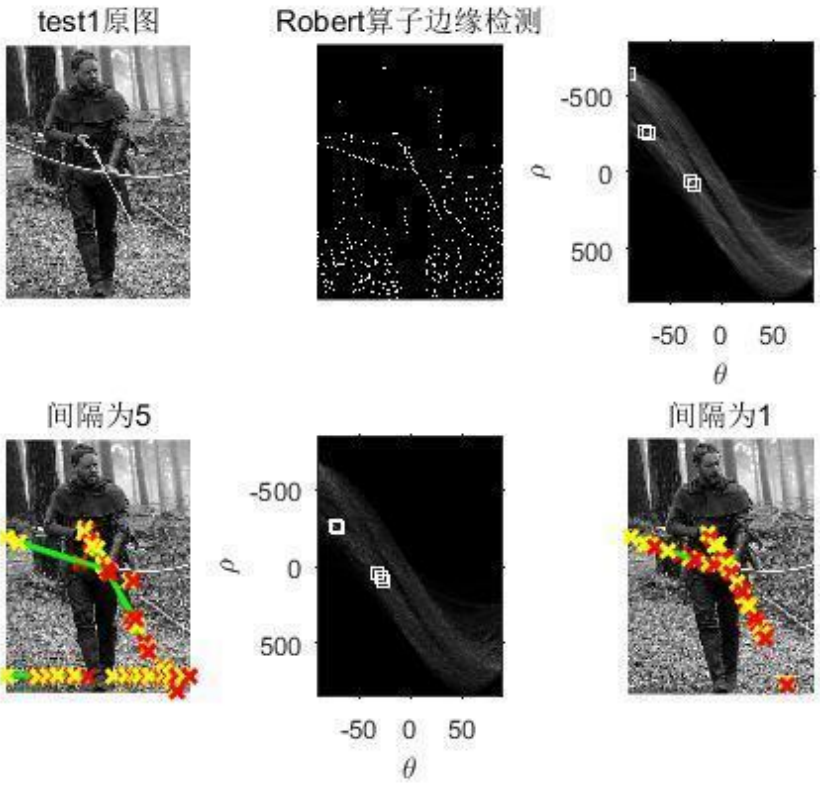
test5：


test1原图
Robert算子边缘检测
-500
0
500
ρ
-50 0 50
θ
间隔为5
间隔为1


test1原图
Sobel算子边缘检测
-500
0
500
ρ
-50
0
50
θ
间隔为5
-500
0
500
ρ
-50
0
50
θ
间隔为1

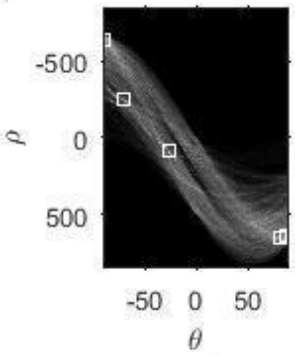

test1原图
Canny算子边缘检测
-500
0
500
ρ
-50 0 50
θ
间隔为5
间隔为1

test6：

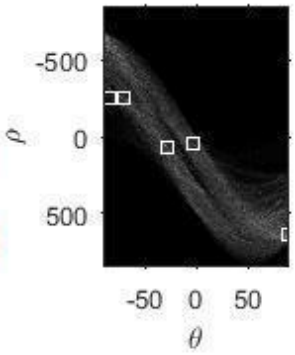

test1原图
Robert算子边缘检测
-500
0
500
ρ
-50 0 50
θ
间隔为5
间隔为1

test1原图

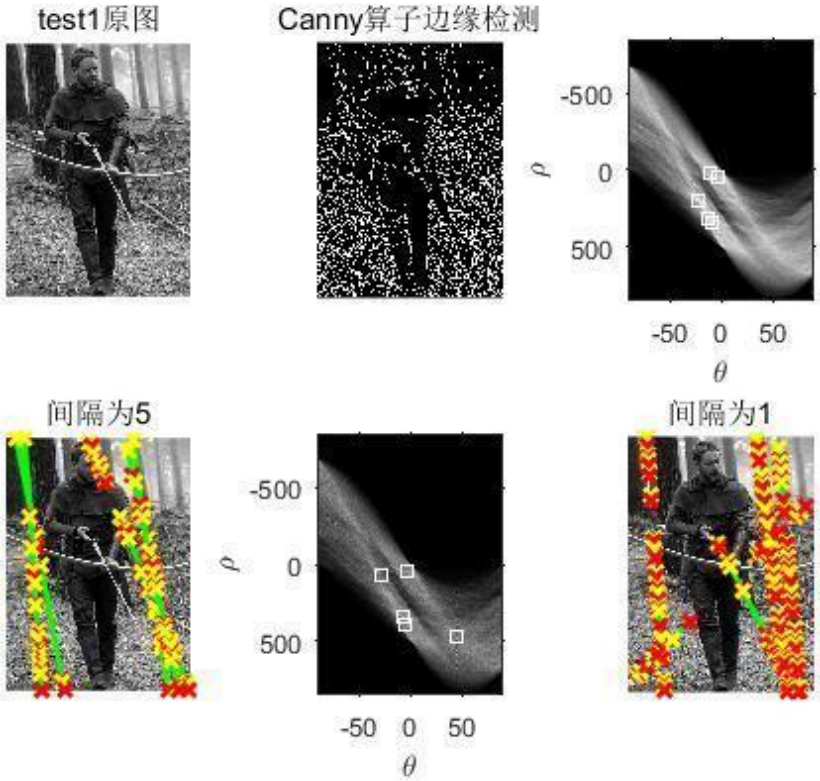
Sobel算子边缘检测



间隔为5



间隔为1



综上可以发现： canny 算子进行边缘检测后，hough 变换后经常会出现一些原图中不存在直线，但观察原图后发现其近似存在一条抽象的直线，Robert 算子得到直线相对较长，而 sobel 算子得到的直线相对较短。 随着间隔的增大，canny 算子，sobel 算子，和 robert 算子都会得到一些抽象的直线，以 test4 为代表，随着间隔增大，hough 变换得到的直线长度有明显增加。

程序代码：

```
im1=imread('test1.tif');

im2=imread('test2.png');

im3=imread('test3.jpg');

im4=imread('test4.bmp');
```

```
im5=imread('test5.png');

im6=imread('test6.jpg');

im1=im1(:,:,1);

a=edge(im1,'canny');

b=edge(im2,'canny');

c=edge(im3,'canny');

d=edge(im4,'canny');

e=edge(im5,'canny');

f=edge(im6,'canny');



figure;

subplot(2,3,1),

imshow(a),axis on;

title('test1');

subplot(2,3,2),

imshow(b),axis on;

title('test2');

subplot(2,3,3),

imshow(c),axis on;

title('test3');

subplot(2,3,4),
```

```
imshow(d),axis on;

title('test4');

subplot(2,3,5),

imshow(e),axis on;

title('test5');

subplot(2,3,6),

imshow(f),axis on;

title('test6');



im=imread('test2.png');



a=edge(im,'sobel',0.02);

b=edge(im,'sobel',0.07);

c=edge(im,'sobel');



figure;

subplot(1,3,1),

imshow(a),axis on;

title('  阈值=0.02');

subplot(1,3,2),

imshow(b),axis on;
```

```
title('  阈值=0.07');

subplot(1,3,3),

imshow(c),axis on;

title('默认');



BW=imread('test2.png');


%BW=rgb2gray(BW);

thresh=[0.01,0.17];

sigma=2;%定义高斯参数

%f = edge(double(BW),'canny',thresh,sigma);

f = edge(double(BW),'canny');

figure(1),imshow(f,[]);

title('canny 边缘检测');


[H, theta, rho]= hough(f,'RhoResolution',0.5);

%imshow(theta,rho,H,[],'notruesize'),axis on,axis normal

%xlabel('\theta'),ylabel('rho');


peak=houghpeaks(H,5);
```

```
hold on



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

figure,imshow(f,[]),title('Hough Transform Detect Result'),hold on

for k=1:length(lines)

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

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

end

im1=imread('test1.tif');

im2=imread('test2.png');

im3=imread('test3.jpg');

im4=imread('test4.bmp');

im5=imread('test5.png');

im6=imread('test6.jpg');

im1=im1(:,:,1);

a=edge(im1,'sobel');

b=edge(im2,'sobel');

c=edge(im3,'sobel');

d=edge(im4,'sobel');

e=edge(im5,'sobel');

f=edge(im6,'sobel');
```

```
figure;
subplot(2,3,1),
imshow(a),axis on;
title('test1');
subplot(2,3,2),
imshow(b),axis on;
title('test2');
subplot(2,3,3),
imshow(c),axis on;
title('test3');
subplot(2,3,4),
imshow(d),axis on;
title('test4');
subplot(2,3,5),
imshow(e),axis on;
title('test5');
subplot(2,3,6),
imshow(f),axis on;
title('test6')
```