数字图像处理第七次作业

姓名：陈泊言

班级：自动化63

学号：2160504056

提交日期：2019/5/14

摘要： 包括本次作业首先通过三种算子的边缘检测来提取边缘像素，再用hough变换检测图中直线并比较不同边缘检测算法（2种以上）和不同hough变换参数对直线检测的影响

一.边缘检测

1.Sobel算子

就是将图像的每一个点都用sobel算子做卷积：一个用来检测垂直边缘，一个用来检测水平边缘，而最后两个卷积的最大值将作为该点的输出，即检测后的灰度。

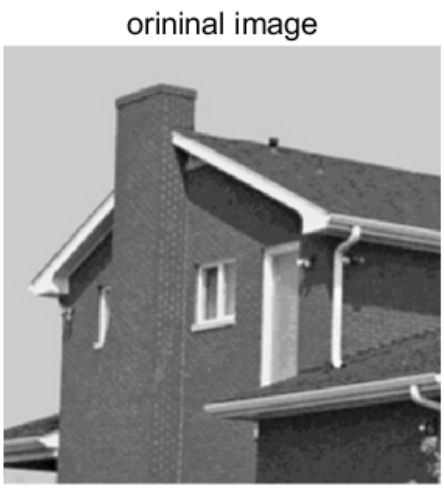
Sobel算子：两组3*3的矩阵，左边的表示垂直，右边的表示水平。将它与图像作平面卷积，即可分别得出垂直及水平的亮度差分近似值。

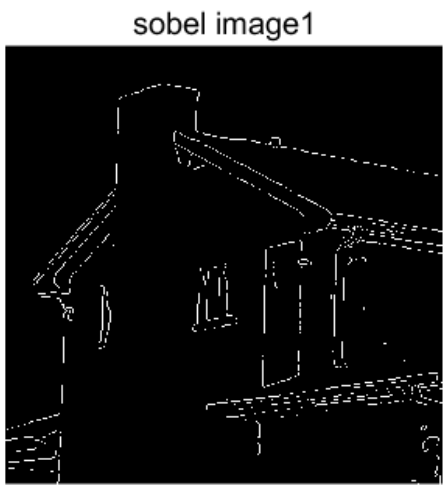
-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

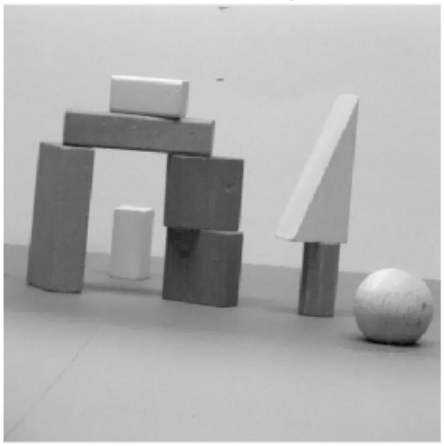
```
i1=imread('test1.tif');
i=i1(:,:,1);
i_can1=edge(i,'sobel');
figure;
subplot(1,2,1),imshow(i),title('orininal image');
subplot(1,2,2),imshow(i_can1),title('sobel image1');

i2=imread('test2.png');
i_can2=edge(i2,'sobel');
figure;
subplot(1,2,1),imshow(i2),title('orininal image');
subplot(1,2,2),imshow(i_can2),title('sobel image2');

i3=imread('test3.jpg');
i_can3=edge(i3,'sobel');
figure;
subplot(1,2,1),imshow(i3),title('orininal image');
subplot(1,2,2),imshow(i_can3),title('sobel image3');
```

```
i4=imread('test4.bmp');
i_can4=edge(i4,'sobel');
figure;
subplot(1,2,1),imshow(i4),title('orininal image');
subplot(1,2,2),imshow(i_can4),title('sobel image4');

i5=imread('test2.png');
i_can5=edge(i5,'sobel');
figure;
subplot(1,2,1),imshow(i5),title('orininal image');
subplot(1,2,2),imshow(i_can5),title('sobel image5');

i6=imread('test6.jpg');
i_can6=edge(i6,'sobel');
figure;
subplot(1,2,1),imshow(i6),title('orininal image');
subplot(1,2,2),imshow(i_can6),title('sobel image6');
```

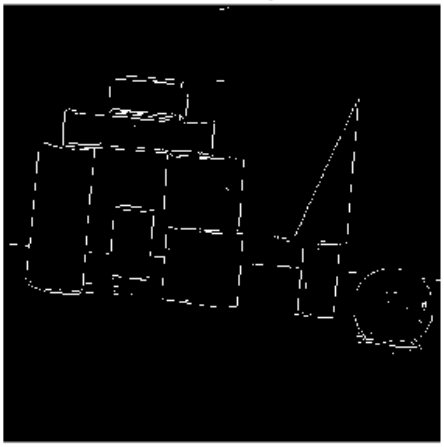




orininal image

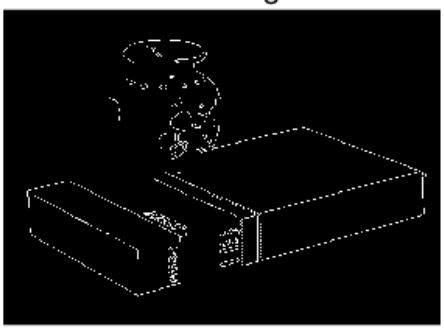


sobel image2

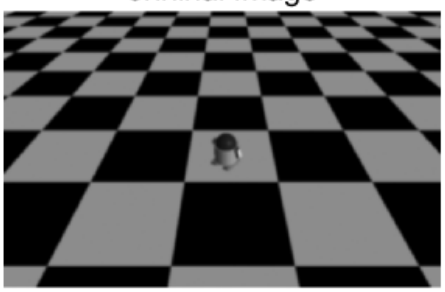


orininal image

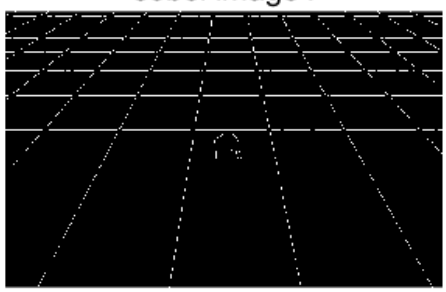
sobel image3



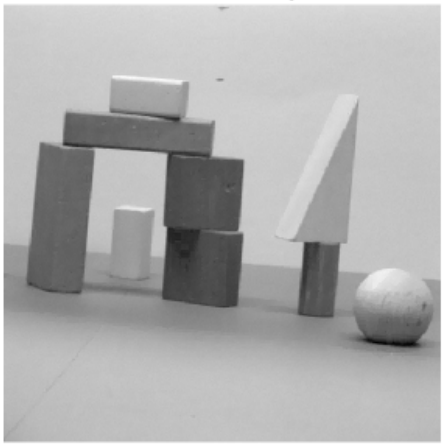
orininal image



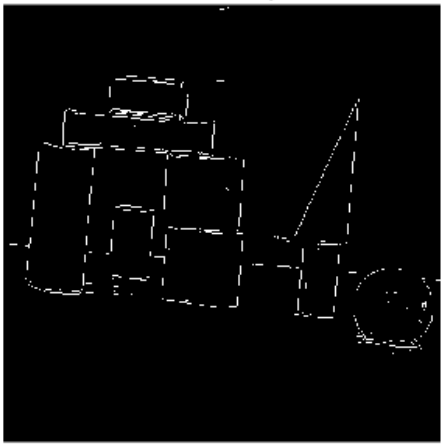
sobel image4



orininal image



sobel image5



sobel 算子是针对特定方向的，对于垂直水平的图像提取效果较好

2.canny:

具体算法步骤：

1:用高斯滤波器平滑图象；

2:用一阶偏导的有限差分来计算梯度的幅值和方向；

3:对梯度幅值进行非极大值抑制；

4:用双阈值算法检测和连接边缘。

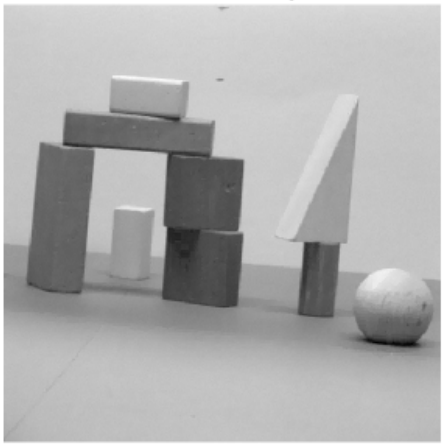
```
i1=imread('test1.tif');
i=i1(:,:,1);
i_can1=edge(i,'canny');
figure;
subplot(1,2,1),imshow(i),title('orininal image');
subplot(1,2,2),imshow(i_can1),title('canny image1');

i2=imread('test2.png');
i_can2=edge(i2,'canny');
figure;
subplot(1,2,1),imshow(i2),title('orininal image');

subplot(1,2,2),imshow(i_can2),title('canny image2');
```

```
i3=imread('test3.jpg');
i_can3=edge(i3,'canny');
figure;
subplot(1,2,1),imshow(i3),title('orininal image');
subplot(1,2,2),imshow(i_can3),title('canny image3');

i4=imread('test4.bmp');
i_can4=edge(i4,'canny');
figure;
subplot(1,2,1),imshow(i4),title('orininal image');
subplot(1,2,2),imshow(i_can4),title('canny image4');

i5=imread('test2.png');
i_can5=edge(i5,'canny');
figure;
subplot(1,2,1),imshow(i5),title('orininal image');
subplot(1,2,2),imshow(i_can5),title('canny image5');

i6=imread('test6.jpg');
i_can6=edge(i6,'canny');
figure;
subplot(1,2,1),imshow(i6),title('orininal image');
subplot(1,2,2),imshow(i_can6),title('canny image6');
```

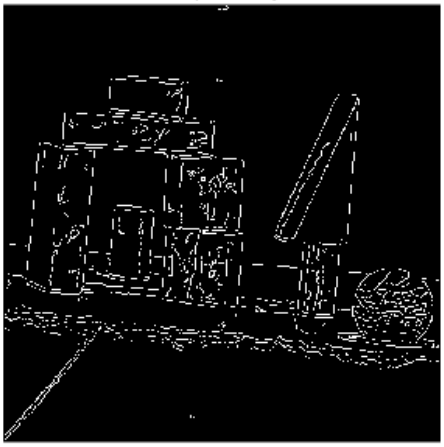
orininal image

canny image1

orininal image

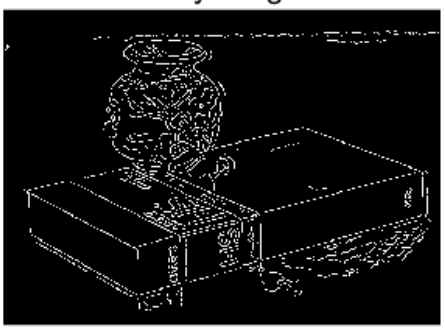


canny image2

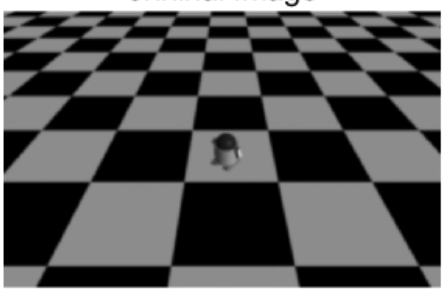


orininal image

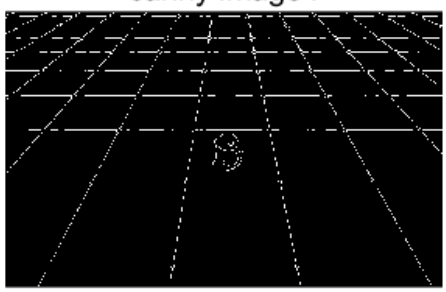
canny image3



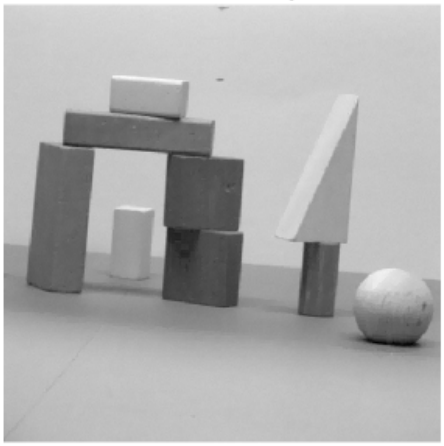
orininal image



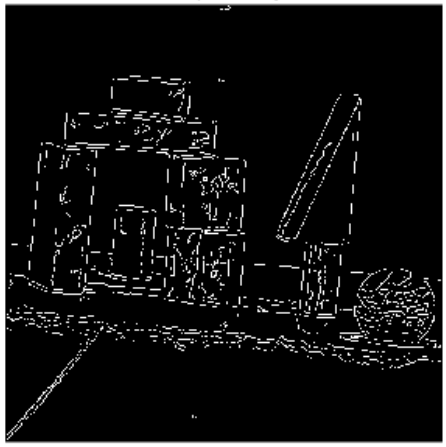
canny image4



orininal image



canny image5



自适应Canny阈值算法提取的细节更多，由于其有高低阈值的设置，因此可以改变阈值再做分析

二.直线检测

我们先通过边缘检测，得到了幅值超过某个阈值的像素集合，即边缘像素。我们想知道，这些边缘像素是否连成直线，即验证这些边缘像素是否在某直线上。为此，我们可以采用**Hough变换。**

原理：

Hough变换通过从直角坐标系到极坐标系的转换，将直角坐标系中的一条＂直线＂，转换为极坐标系上的一个＂点＂，落在这条＂直线＂上的像素点越多，这个极坐标中＂点＂的权越重，最终通过分析各个＂点＂的权重（局部最大值）v

采用两组参数，ρ 和θ 。其中ρ 代表由原点(一般是图像中心)到直线的距离，θ 代表直线的倾角。在这个参数空间里，过一点A的所有直线映射到参数空间会形成一条正弦曲线。同样的，交点代表同时过几个点的直线，相交于同一点的曲线越多，说明这组参数所代表的直线越显著。

matlab提供了三个与霍夫变换有关的函数。函数hough实现了前面讨论的概念，函数houghpeaks寻找霍夫变换的峰值(累加单元的高计数)，函数houghlines以来自其他两个函数的结果为基础在原始图像中提取线段。

1.函数hough

H, theta, rho] = hough(f, 'ThetaRes', val1, 'RhoRes', val2)

其中，H是霍夫变换矩阵，theta(以度计)和rho是ρ和θ值向量，在这些值上产生霍夫变换。

2.函数houghpeaks

完整的语法形式：peaks = houghpeaks(..., 'Threshold', val1, 'NHoodSize', val2)

其中，"..."指出来自默认语法和peaks的输入是持有峰值行和列坐标的Q×2大小的矩阵。Q的范围是0到NumPeaks，H是霍夫变换矩阵。

这个过程的基本思想是：通过把发现峰值的直接邻域中的霍夫变换单元置0来清理峰值。

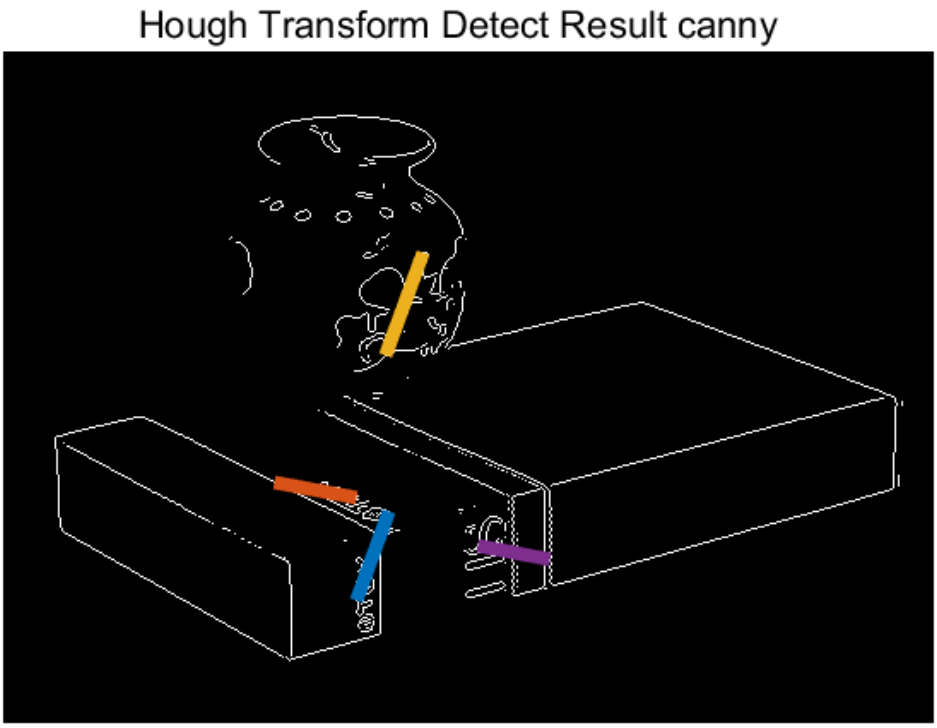
3.函数houghlines

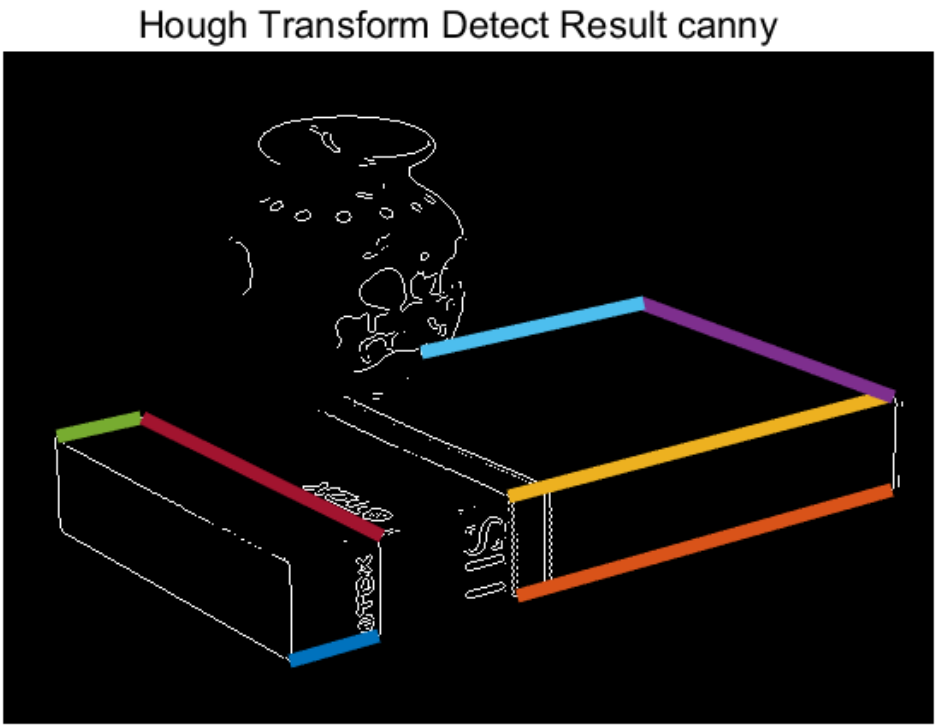
一旦一组候选的峰值在霍夫变换中被识别出来，如果存在与这些峰值相关的有意义的线段，剩下的就是决定线的起始点和终点

lines = houghlines(f, theta, rho, peaks)

代码举例，以canny算法为例

```
i1=imread('test3.jpg');
i=i1(:,:,1);
BW=edge(i,'canny');
figure;
subplot(1,2,1),imshow(i),title('orininal image');
subplot(1,2,2),imshow(BW),title('canny image1');
[H, theta, rho]= hough(BW,'Theta',-80:100:80);
peak=houghpeaks(H,5);
lines=houghlines(BW,theta,rho,peak);
figure,imshow(BW,[]),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);
end

i2=imread('test3.jpg');
BW=edge(i2,'canny');
figure;
subplot(1,2,1),imshow(i2),title('orininal image');
subplot(1,2,2),imshow(BW),title('canny image2');
[H, theta, rho]= hough(BW,'Theta',-50:0.1:50);
peak=houghpeaks(H,5); %求极值点
lines=houghlines(BW,theta,rho,peak); %返回原图直线信息
figure,imshow(BW,[]),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);
end
```


Hough Transform Detect Result canny


Hough Transform Detect Result canny

三.结果分析

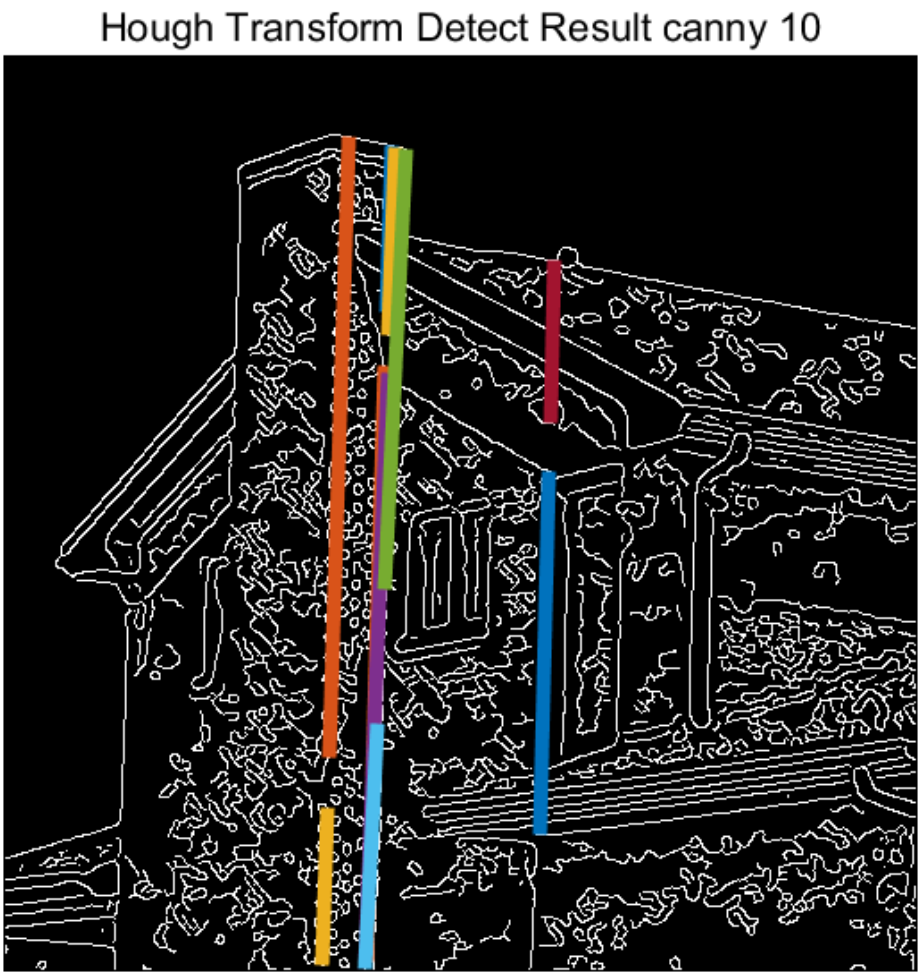
边缘检测总体效果canny算子最佳，sobel 检测较粗，对于垂直和水平的轮廓识别明显

直线检测很大程度取决与参数的选择，对于不同的效果，需要不断调整到合适的参数

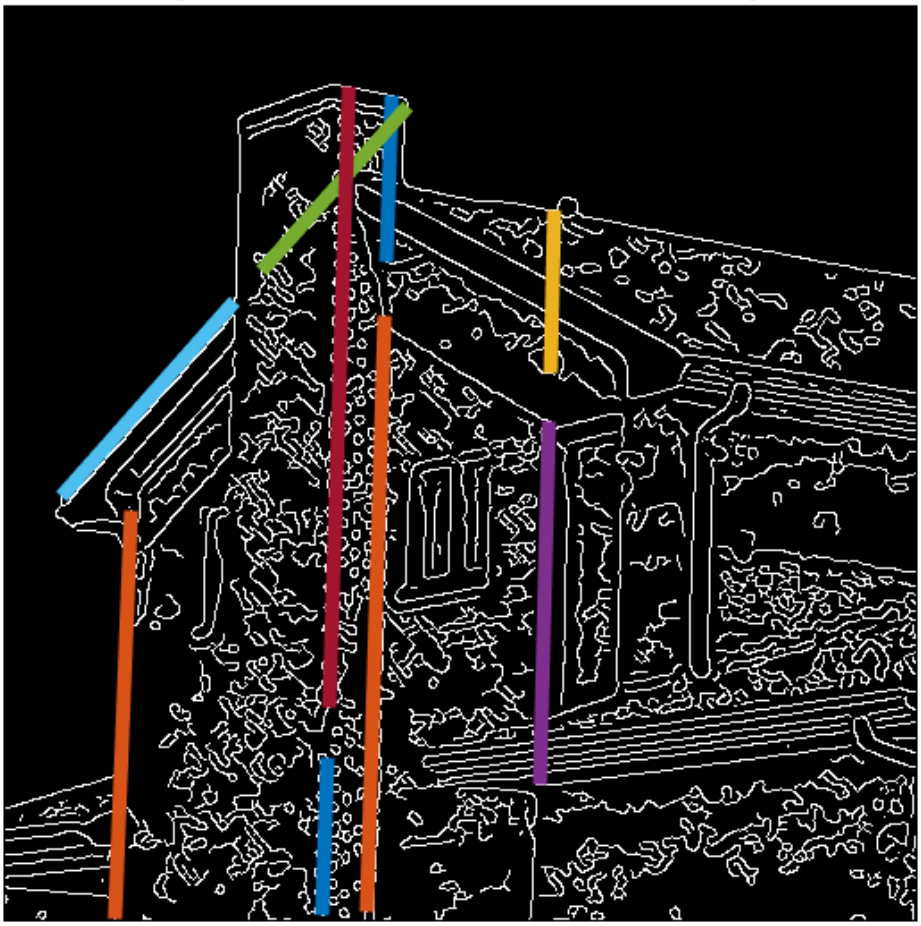
1.改变theta得到四张图，test1绝对值分别为10，50，80，test2 80

当参数选为10 只能检测出垂直的直线

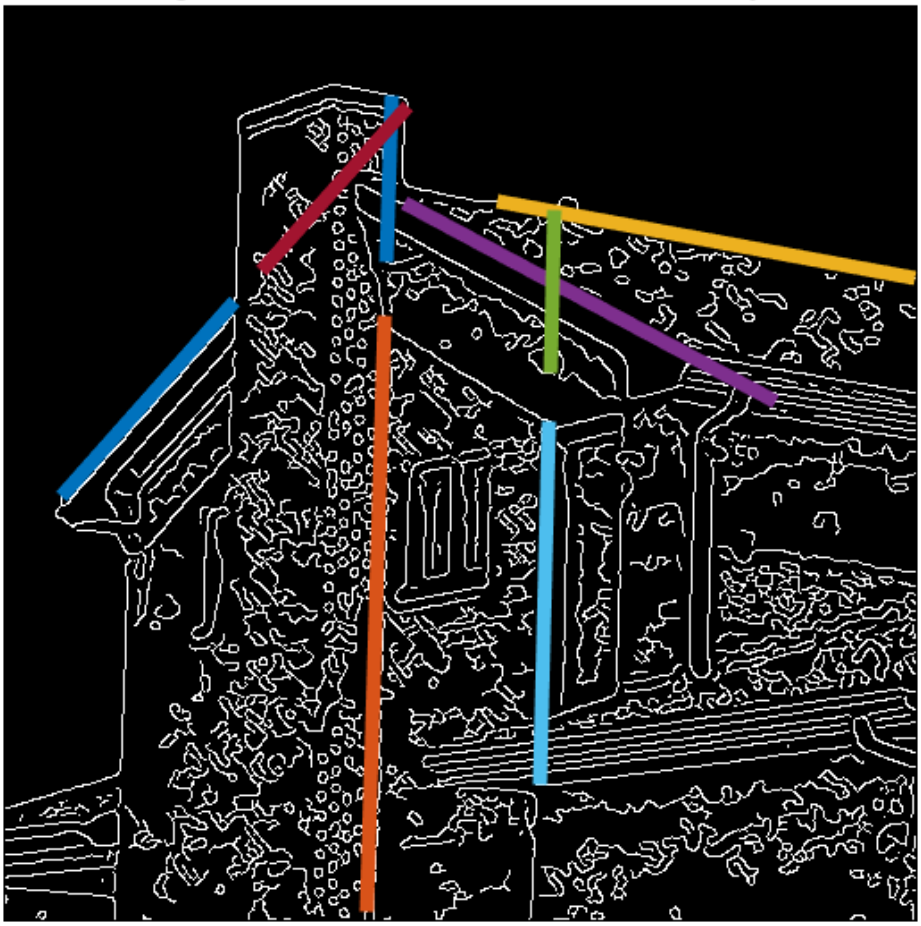
为50时可以检测出斜线，而当参数取80时近乎只能检测出水平的线，由于test1水平线较少，从test2的结果可以很明显的看出

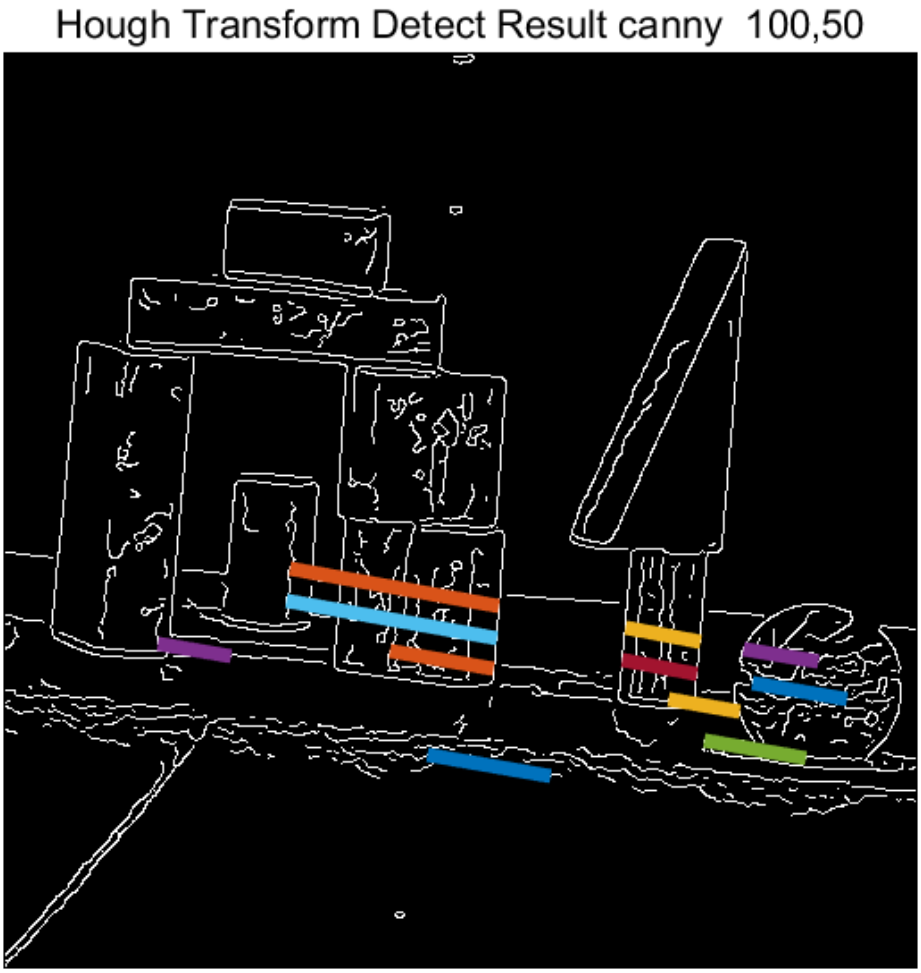


Hough Transform Detect Result canny 50



Hough Transform Detect Result canny 90





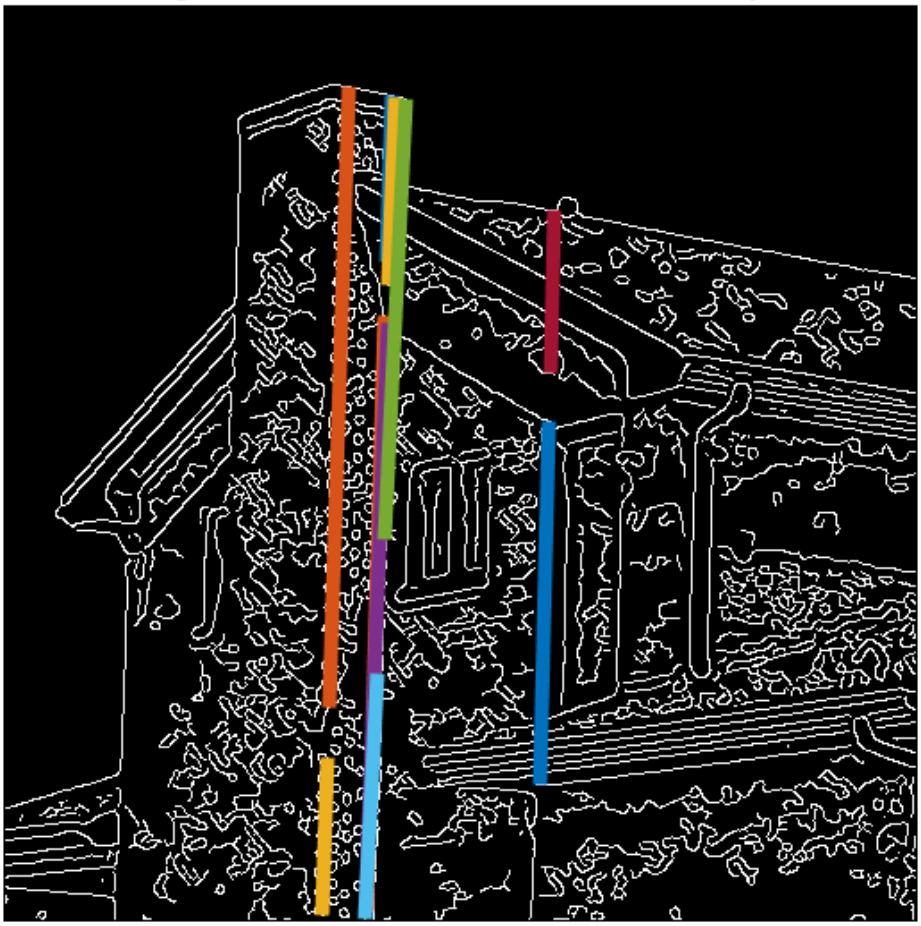
2.改变rho

图为rho增大，参数分别为0.1 10 50，theta保持为10不变，观察垂直直线

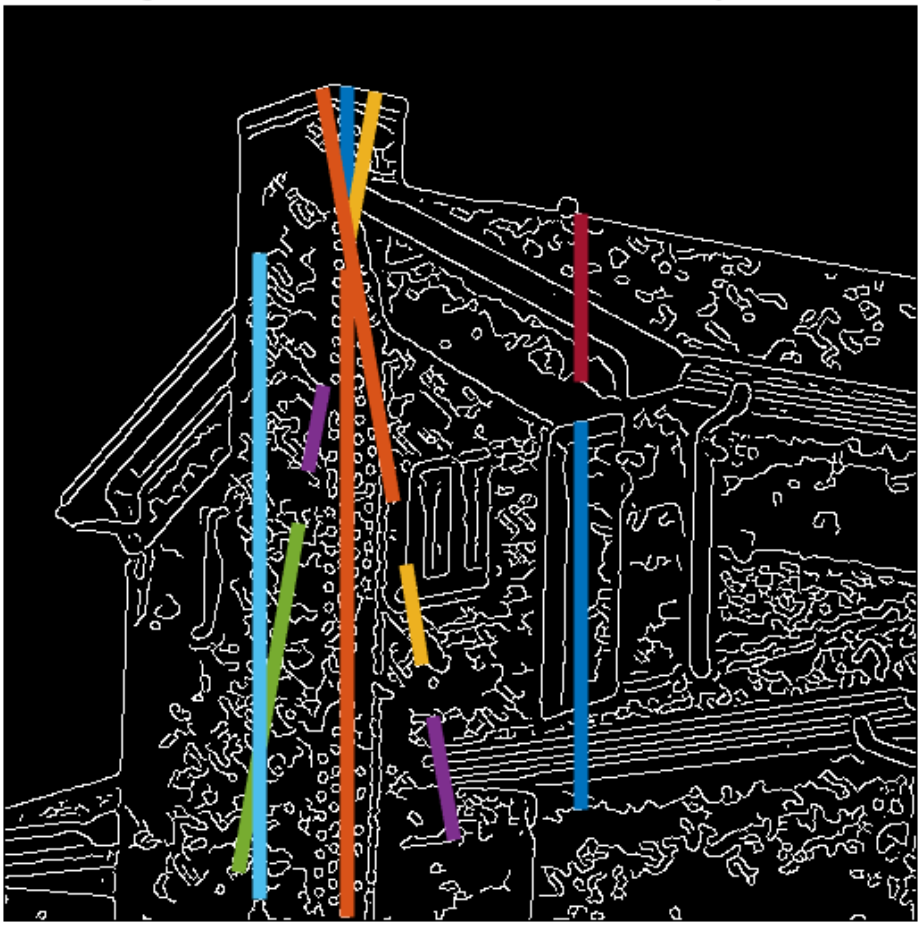
对于细节要求变低后，使得出现了不是直线的被误认为直线提取出来，同样，要根据具体的图分析，test1不适合更大的细分

对比多张图，rho参数不应选取过大，会出现伪直线检测，比如test4中

Hough Transform Detect Result canny 10



Hough Transform Detect Result canny 10,10



Hough Transform Detect Result canny 10,50