直线检测

自动化 62　　2160300167　　张斐然

实验要求

1. 首先对测试图像（文件名为：test1~test6）进行边缘检测，可采用书上介绍的 Sobel 等模板或者 cann 算子方法；
2. 在边缘检测的基础上，用 hough 变换检测图中直线；
3. 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；
4. 可以采用 Matlab、OpenCV 等自带函数。

实验原理

图像空间是所有像素所属于的图像的空间。Hough 空间是一种变量混合空间，实际上它与图像相关但是却不存在物理实质性。

我们可以把图像空间的坐标通过下式表达成 Hough 空间：

$X = P \cdot \cos\Theta$

$Y = P \cdot \sin\Theta$

where

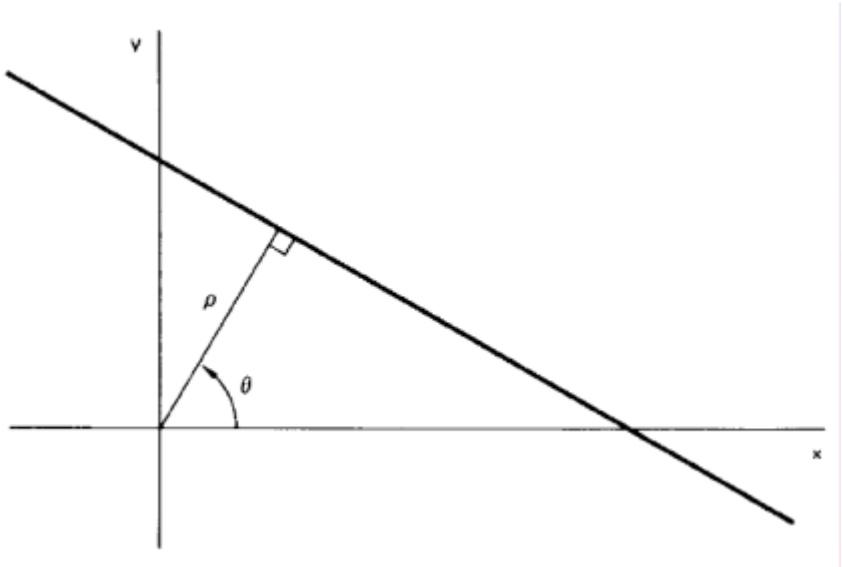
$P = \sqrt{x^2+y^2}$，是坐标原点到直线的距离

$\theta = arctan(\frac{x}{y}), (x,y) \in \mathbb{R}^{+2}$，是距离与 x 坐标轴的夹角

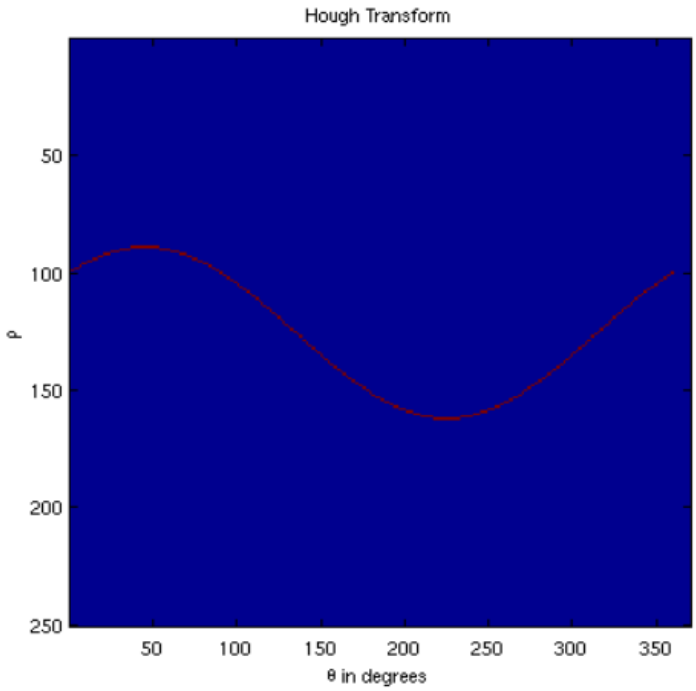
通常我们写成如下形式：

$$\rho = xcos(\theta) + ysin(\theta)$$

通过下图我们可以更加容易理解上述式子：

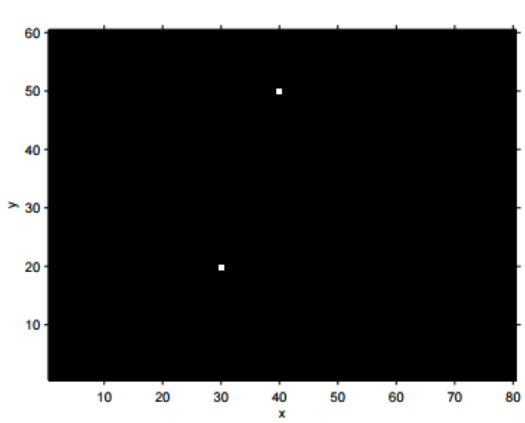


经过 Hough 变换我们将图像空间中的一个点映射到 Hough 空间，如下图我们得到了一条正弦曲线。

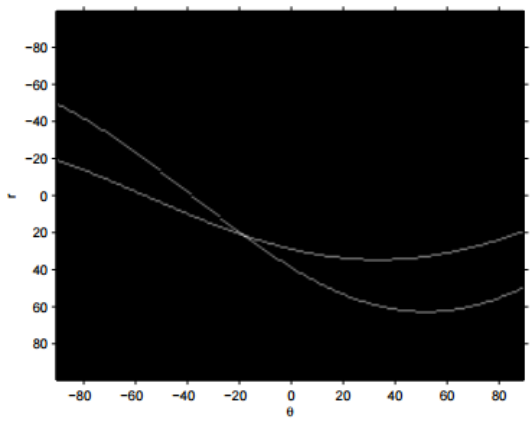


在这里正弦曲线的形状取决于，点到我们所定义原点的距离。通常，距离越大，正弦曲线的振幅越大，反之则会变小。为了使曲线显示我们把纵坐标设置成如上，当然我也可以用 π 表示。

以同样的方法我们可以再次映射一个点，而我们知道在图像空间中两个点总在一条直线上。而在 Hough 空间中我们可以看到两条正弦曲线可能会相交如下图：



(a) Points p_0 and p_1.



(b) All possible lines through p_0 and/or p_1 represented in the Hough space.

在这里我们可以把每一个交点看成是一次投票，也就是

$$\mathcal{A}(\hat{\rho},\hat{\theta}) = \mathcal{A}(\hat{\rho},\hat{\theta}) + 1$$

计算完所有边缘点后，我们可以设置一个阈值，投票大于这个阈值的点这是我们要找的直线。

对于大于阈值的点我们有其 Hough space 的参数对(p,Θ)，通过逆映射我们可以得到图像空间中的直线：

$$y = \frac{-cos(\theta_l)}{sin(\theta_l)}x + \frac{\rho_l}{sin(\theta_l)}$$

实验内容

1. 首先对测试图像（文件名为：test1~test6）进行边缘检测，可采用书上介绍的 Sobel 等模板或者 cann 算子方法；

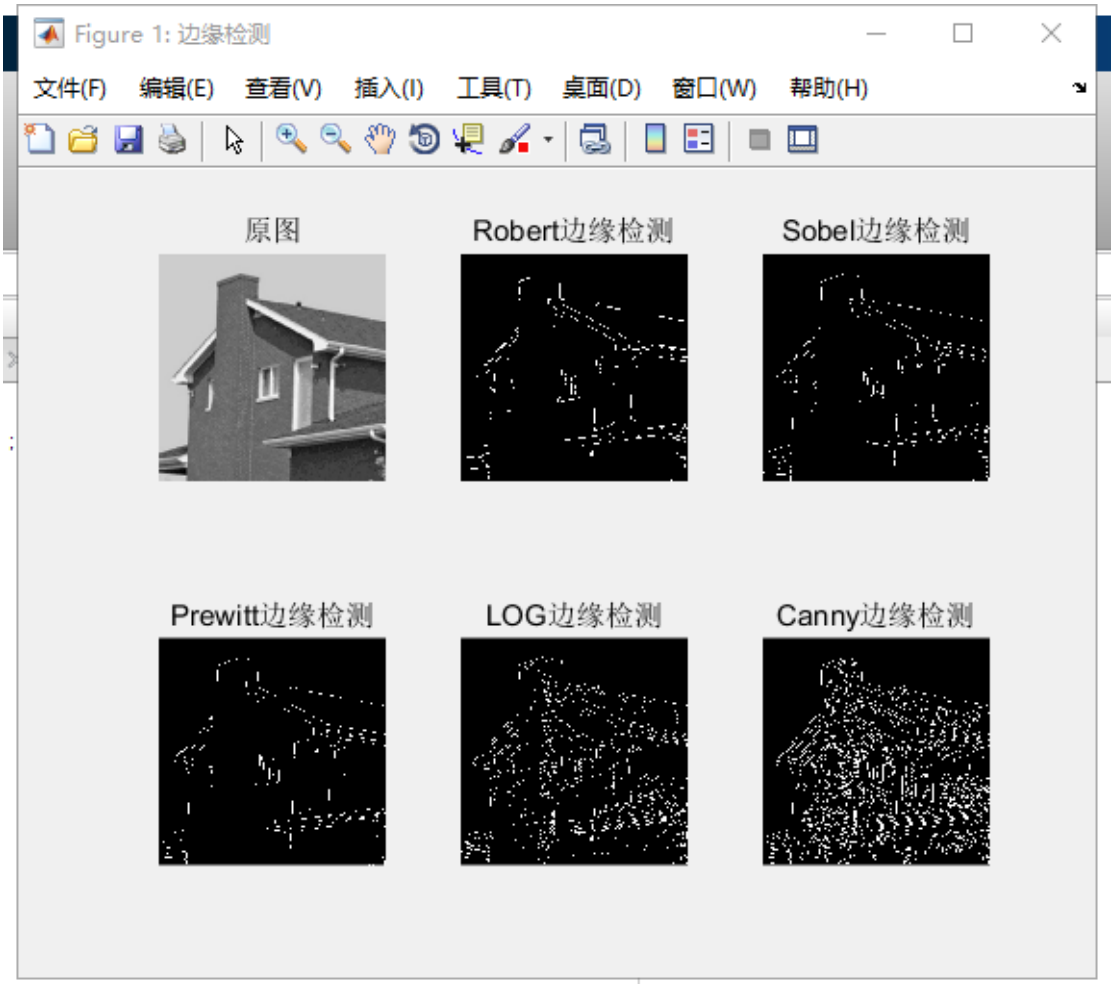


Figure 1: 边缘检测
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原图
Robert边缘检测
Sobel边缘检测
Prewitt边缘检测
LOG边缘检测
Canny边缘检测

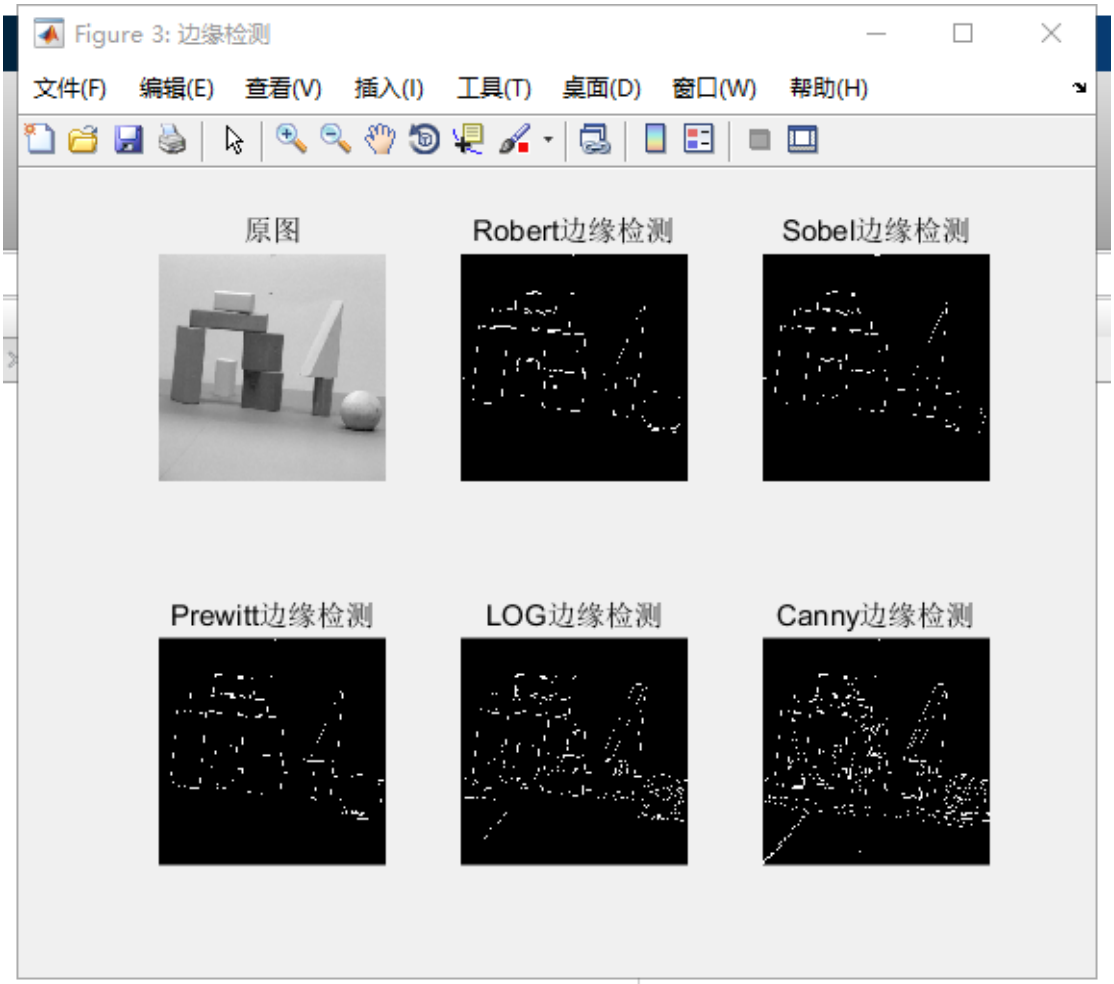


Figure 3: 边缘检测
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原图
Robert边缘检测
Sobel边缘检测
Prewitt边缘检测
LOG边缘检测
Canny边缘检测

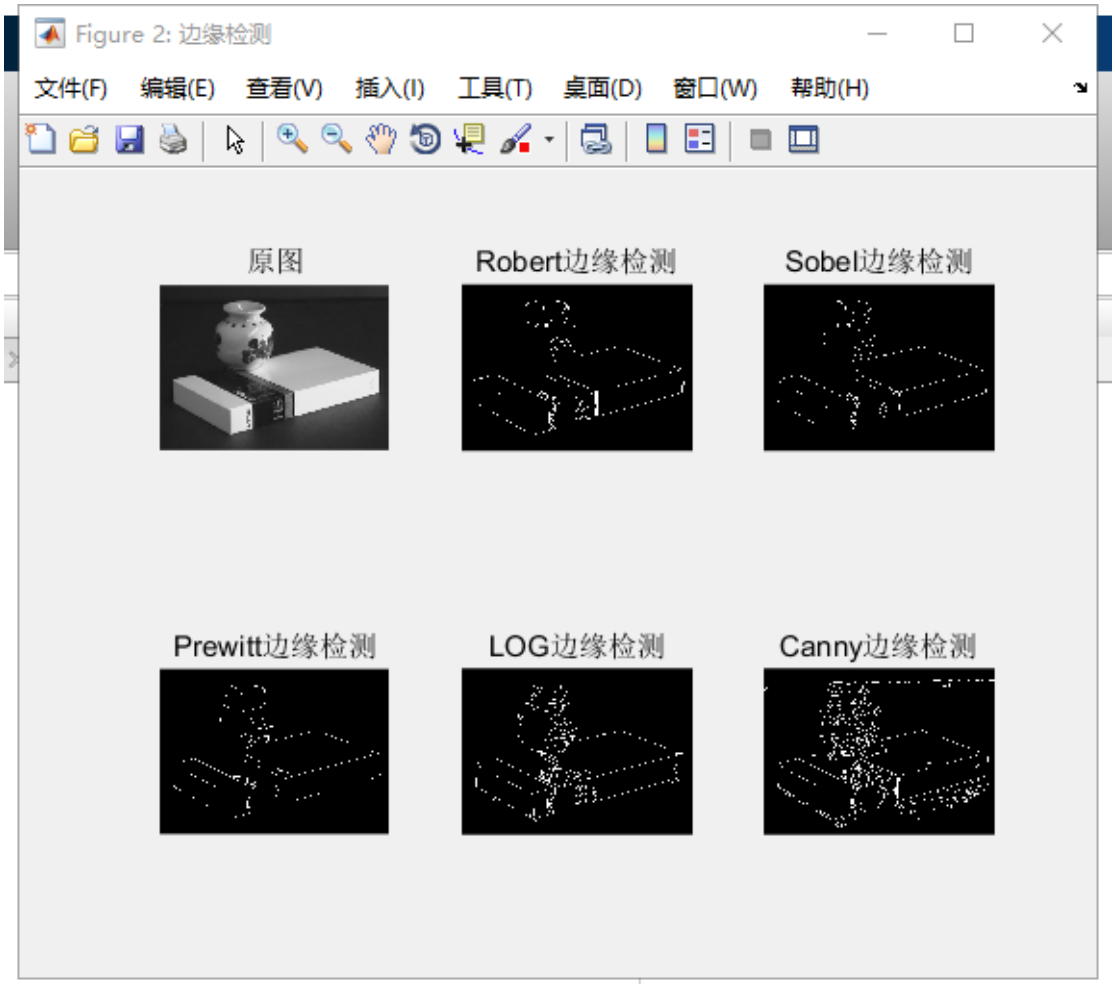


Figure 2: 边缘检测
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原图
Robert边缘检测
Sobel边缘检测
Prewitt边缘检测
LOG边缘检测
Canny边缘检测

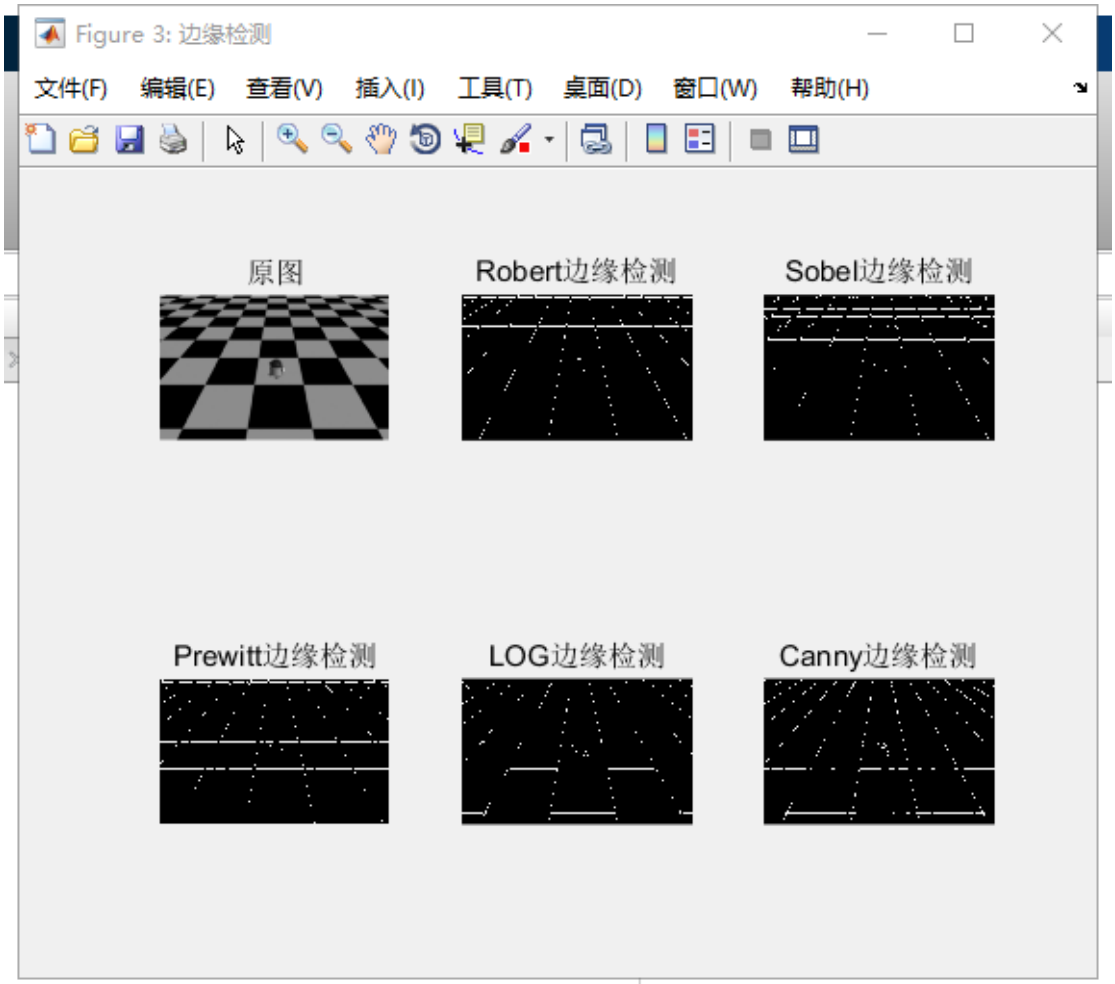


Figure 3: 边缘检测
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
原图
Robert边缘检测
Sobel边缘检测
Prewitt边缘检测
LOG边缘检测
Canny边缘检测

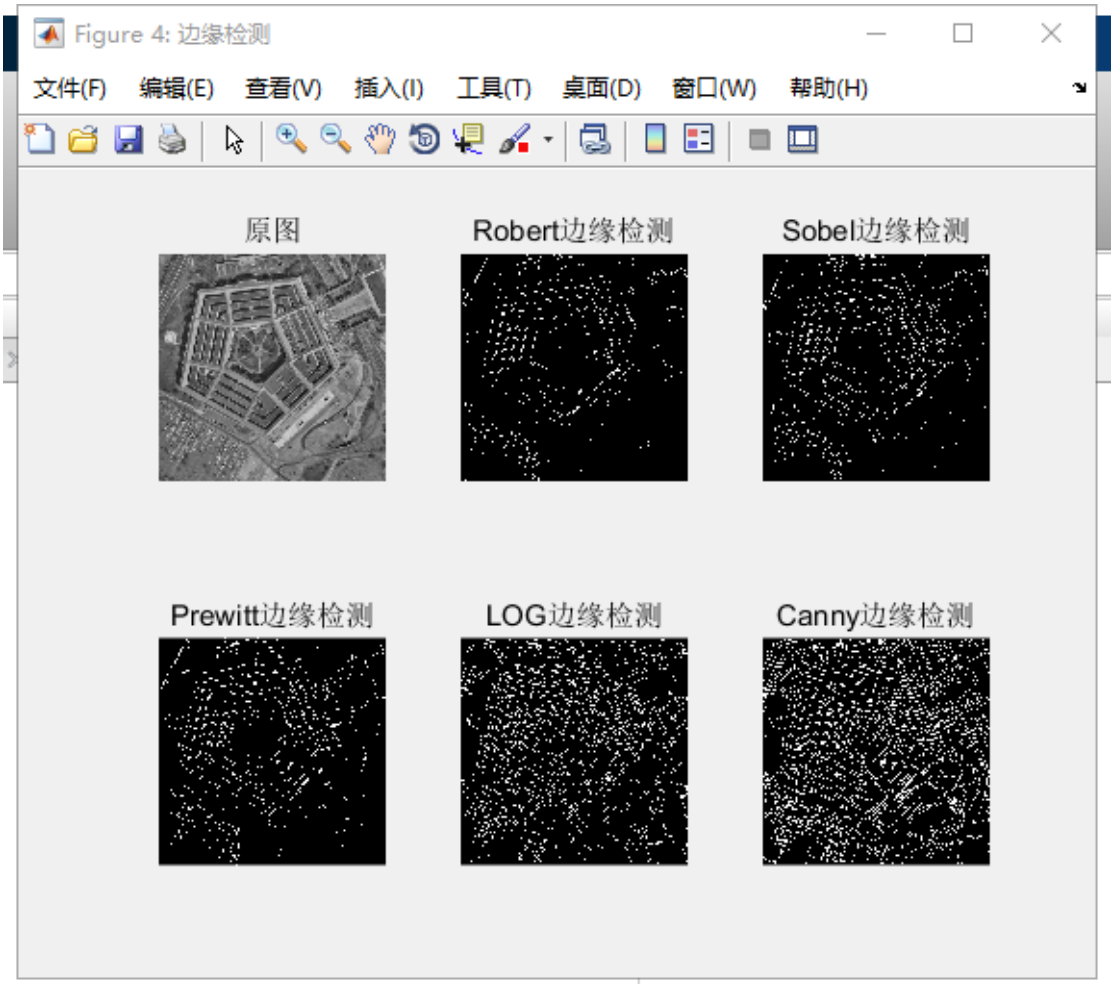
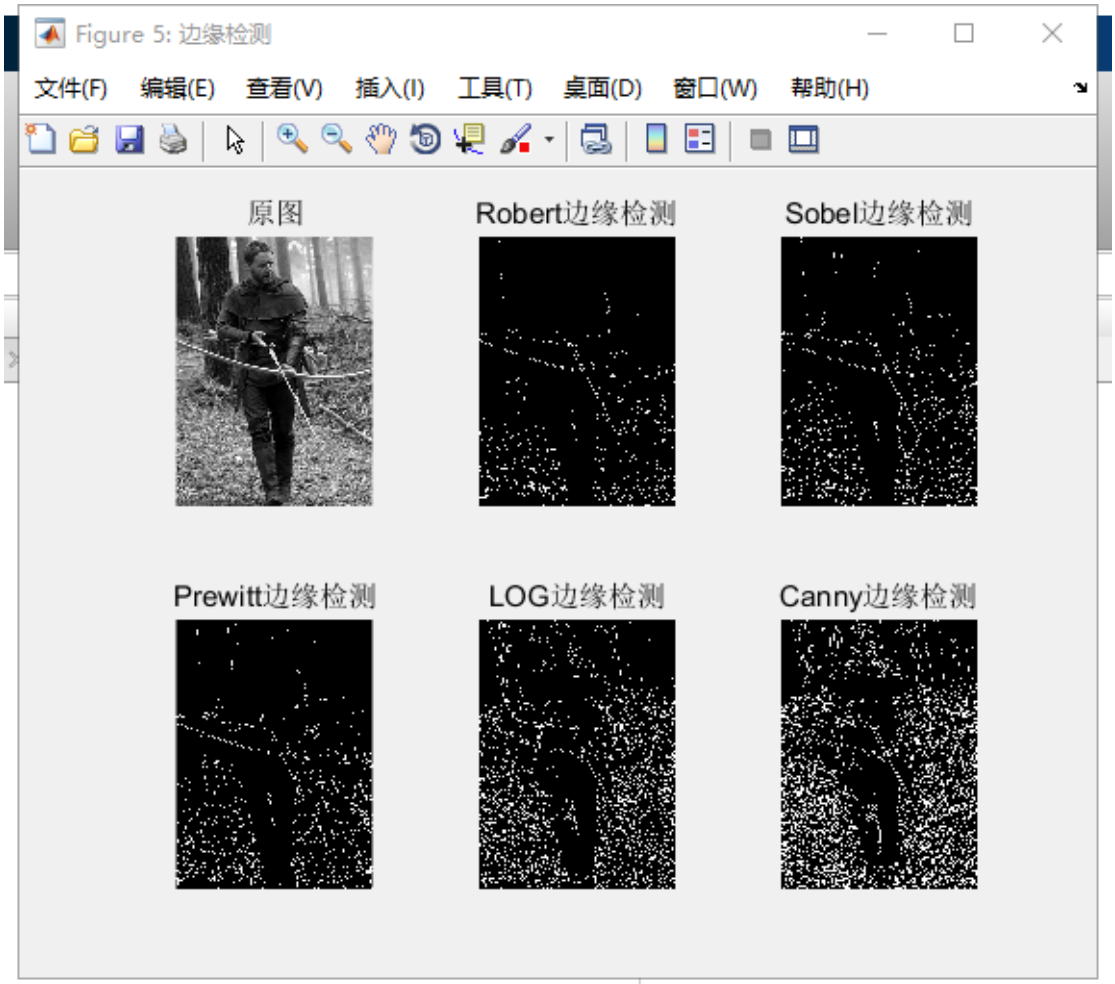


Figure 4: 边缘检测
文件(F)
编辑(E)
查看(V)
插入(I)
工具(T)
桌面(D)
窗口(W)
帮助(H)
原图
Robert边缘检测
Sobel边缘检测
Prewitt边缘检测
LOG边缘检测
Canny边缘检测



2. 在边缘检测的基础上，用 hough 变换检测图中直线；

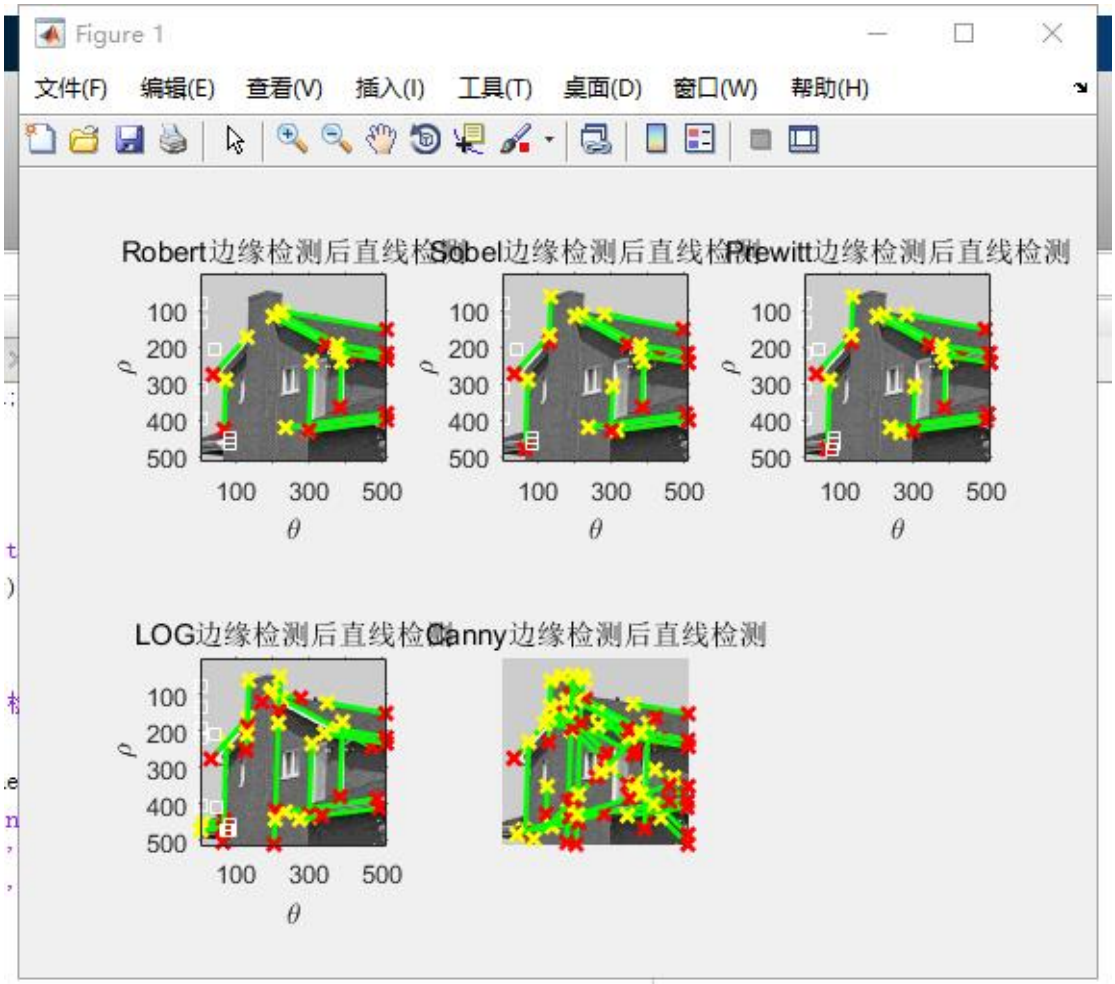


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Robert边缘检测后直线检测
Sobel边缘检测后直线检测
Prewitt边缘检测后直线检测
LOG边缘检测后直线检测
Canny边缘检测后直线检测
ρ
θ
100
200
300
400
500
100
300
500

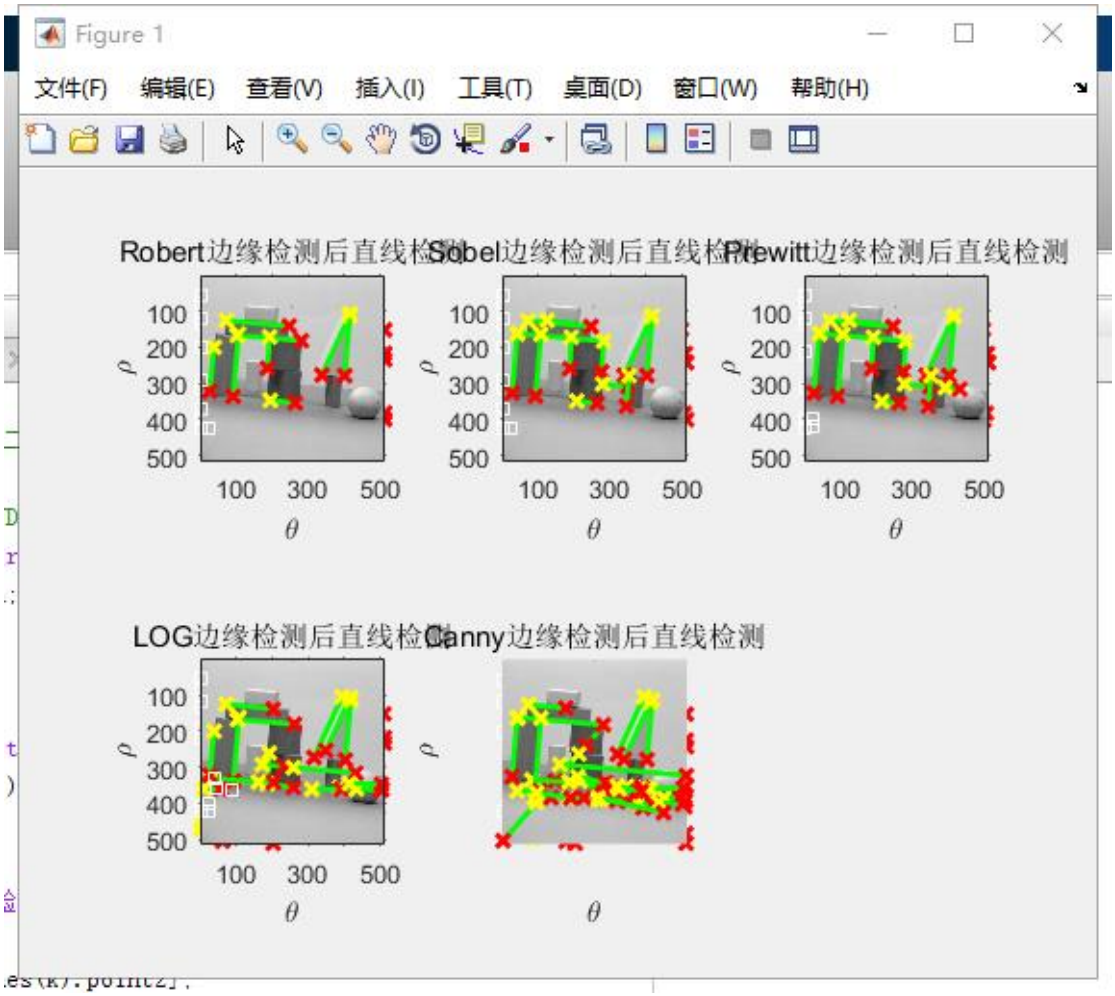


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Robert边缘检测后直线检测
Sobel边缘检测后直线检测
Prewitt边缘检测后直线检测
LOG边缘检测后直线检测
Canny边缘检测后直线检测
ρ
θ
100
200
300
400
500

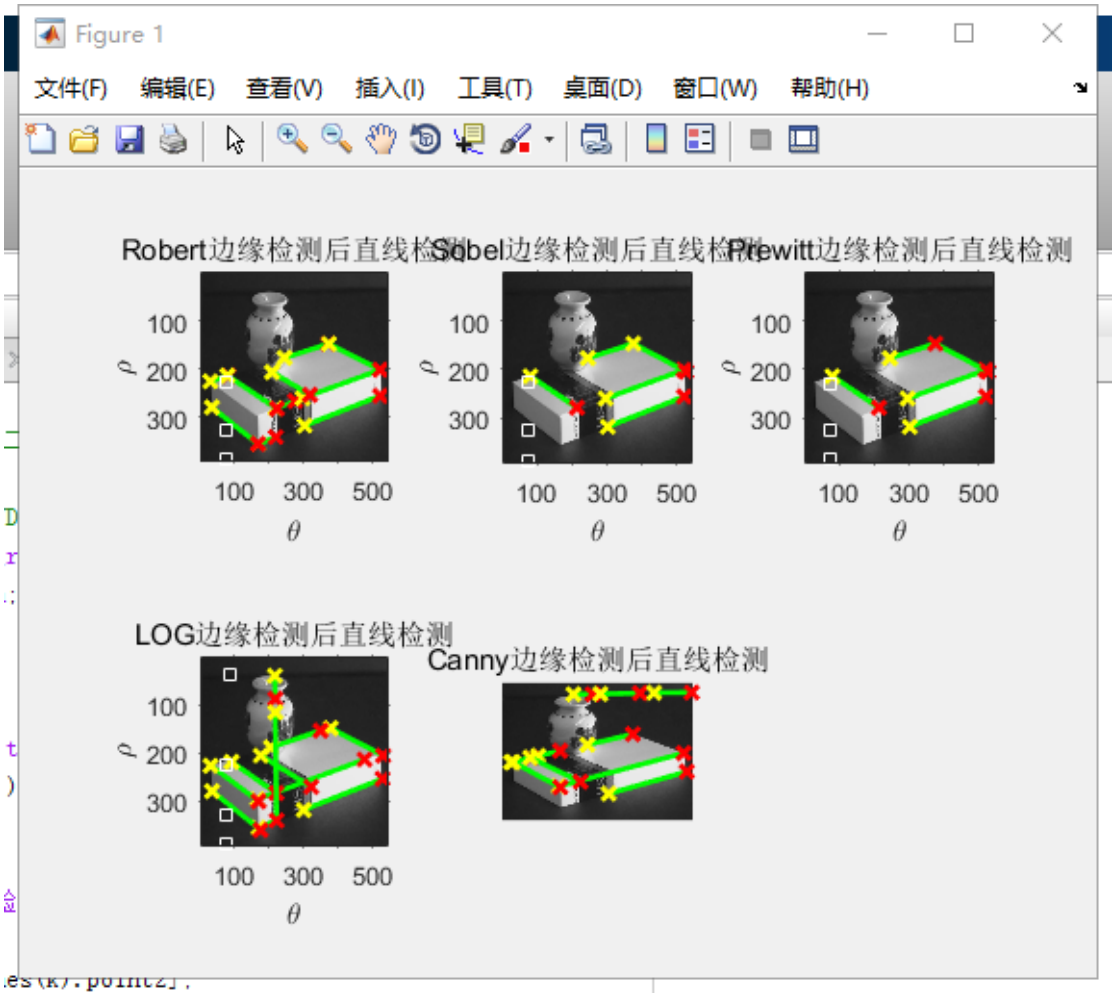


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Robert边缘检测后直线检测
Sobel边缘检测后直线检测
Prewitt边缘检测后直线检测
LOG边缘检测后直线检测
Canny边缘检测后直线检测
ρ
θ
100
200
300
100
300
500

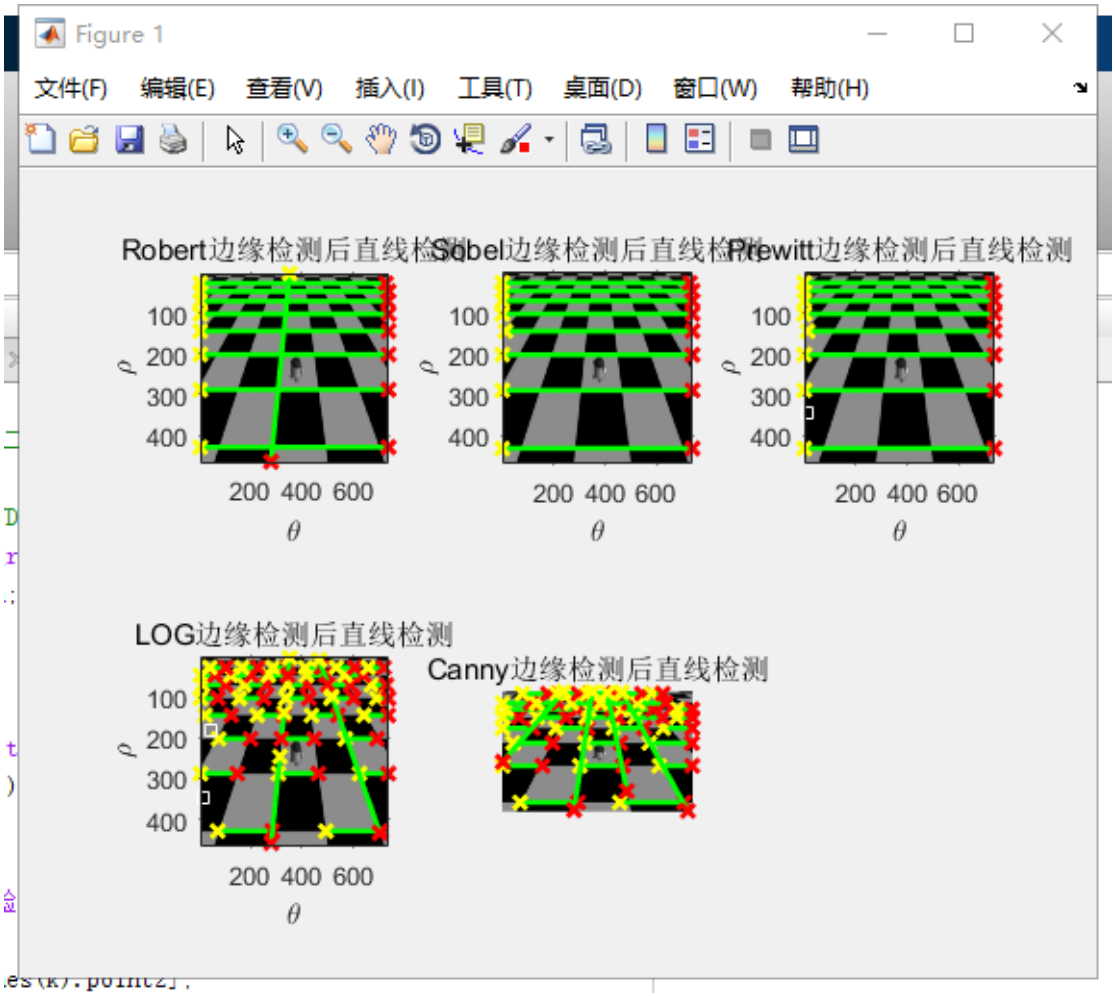


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Robert边缘检测后直线检测
Sobel边缘检测后直线检测
Prewitt边缘检测后直线检测
LOG边缘检测后直线检测
Canny边缘检测后直线检测
100
200
300
400
ρ
200 400 600
θ

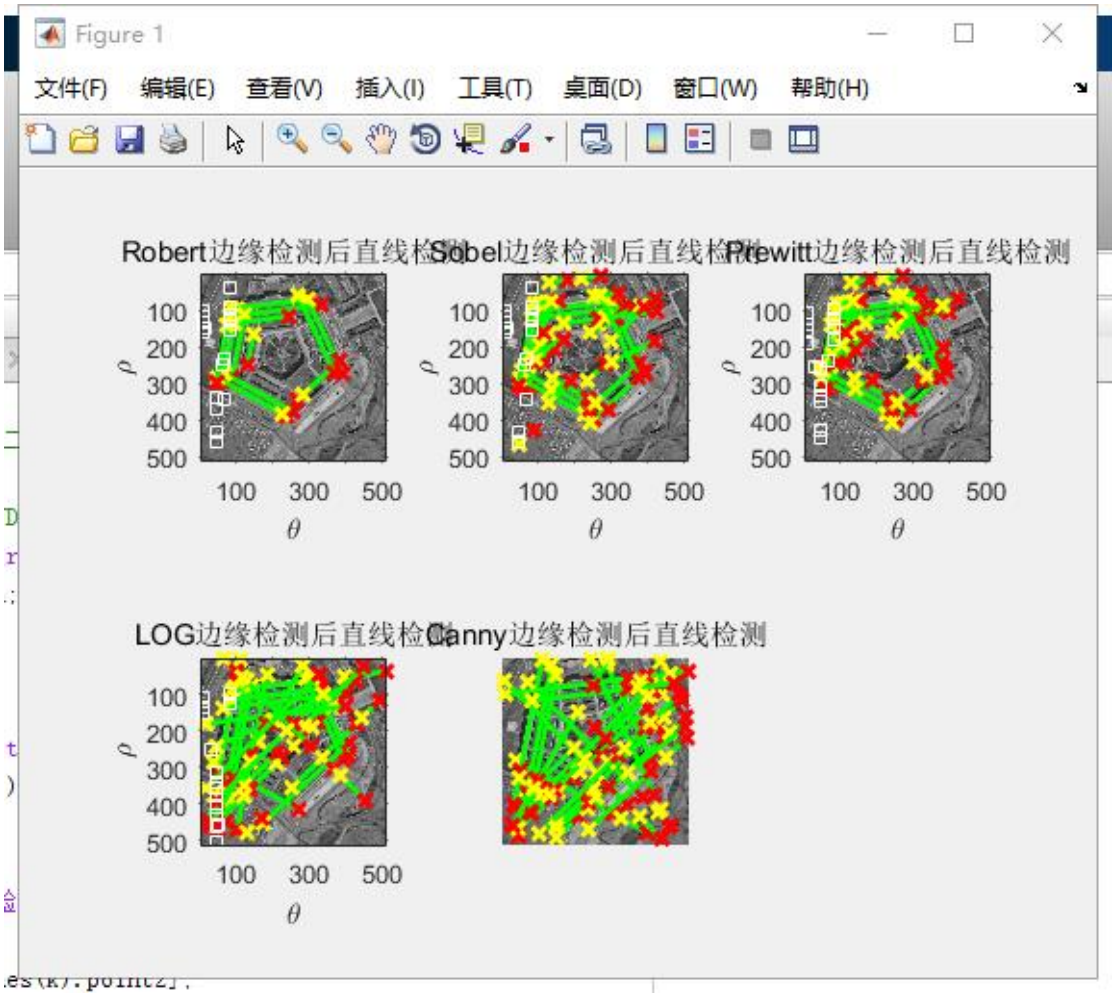
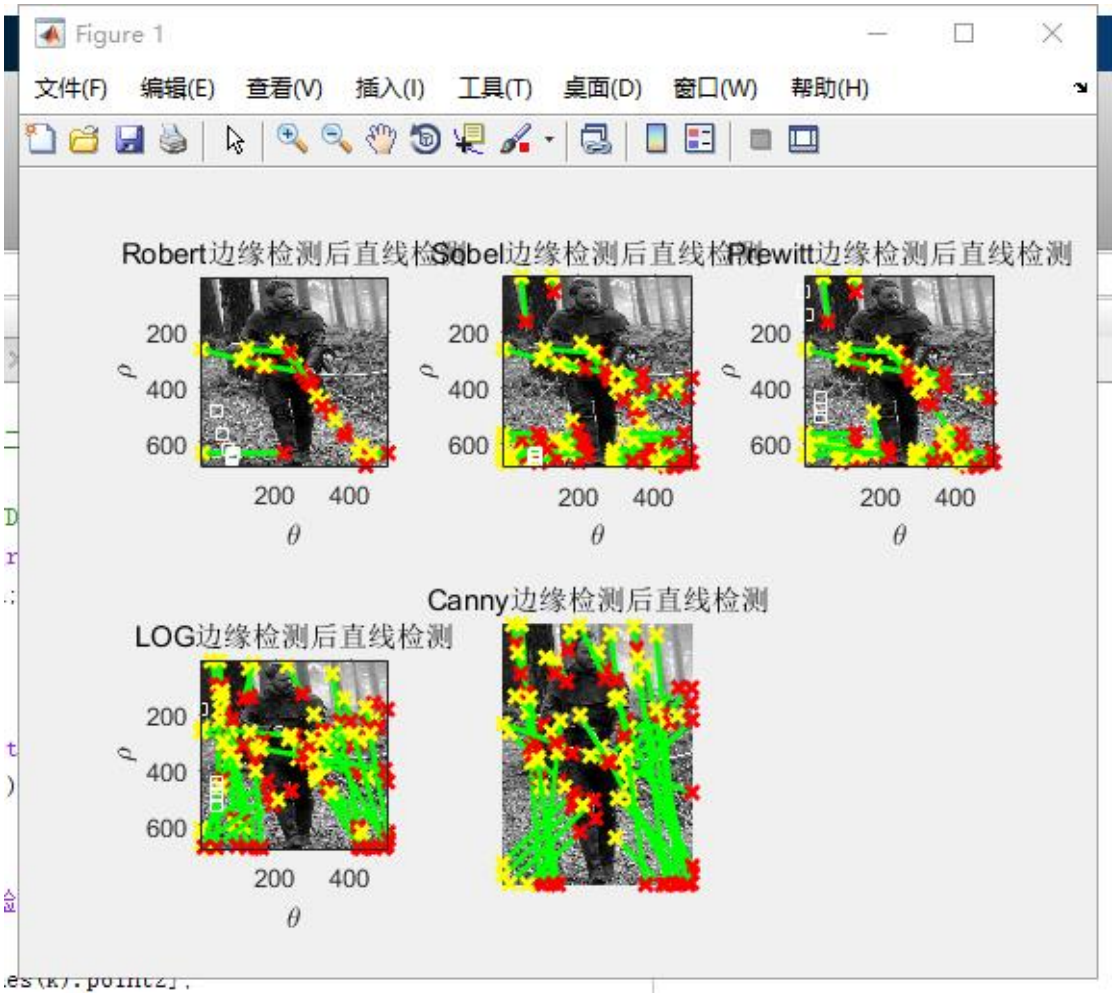


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Robert边缘检测后直线检测
Sobel边缘检测后直线检测
Prewitt边缘检测后直线检测
LOG边缘检测后直线检测
Canny边缘检测后直线检测
ρ
θ
100
200
300
400
500
100
300
500



结论：

通过以上几幅图像我们可以看出，Robert 算子由于边缘检测效果最差，所以之间检测后检测出的直线最少。相对的，由于 canny 算子边缘检测能力最好，所以直线检测后检测出的直线数目较多。但是应该注意比如 test2，由于 canny 边缘检测效果较好，导致了直线检测检测出我们不希望的线段，比如将地上的阴影也检测出了直线，这时采用前四种边缘检测方法较好。但是对于 test3，图片构造比较简单时，使用 LOG 或者 canny 算子可以提高检测的准确性。再比如 test6，地上的树枝实际上也构成了数量极其多的直线，如果我们想检测出这种直线，我们需要用后两种算子。如果不需要，我们可以只采用 Roberts 算子。

3. 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；

改变 Rhoresolution 值：

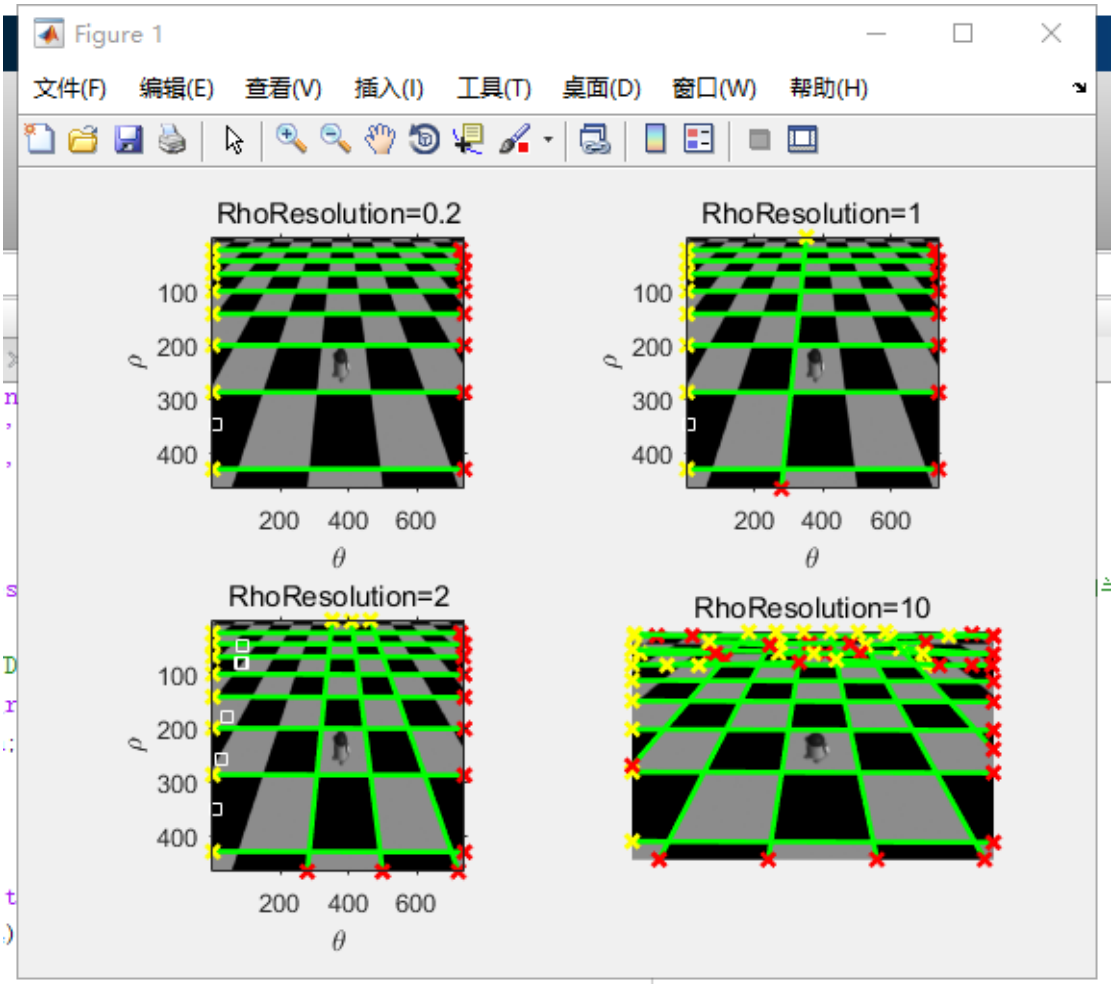


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
RhoResolution=0.2
RhoResolution=1
RhoResolution=2
RhoResolution=10
ρ
θ
100
200
300
400
200
400
600

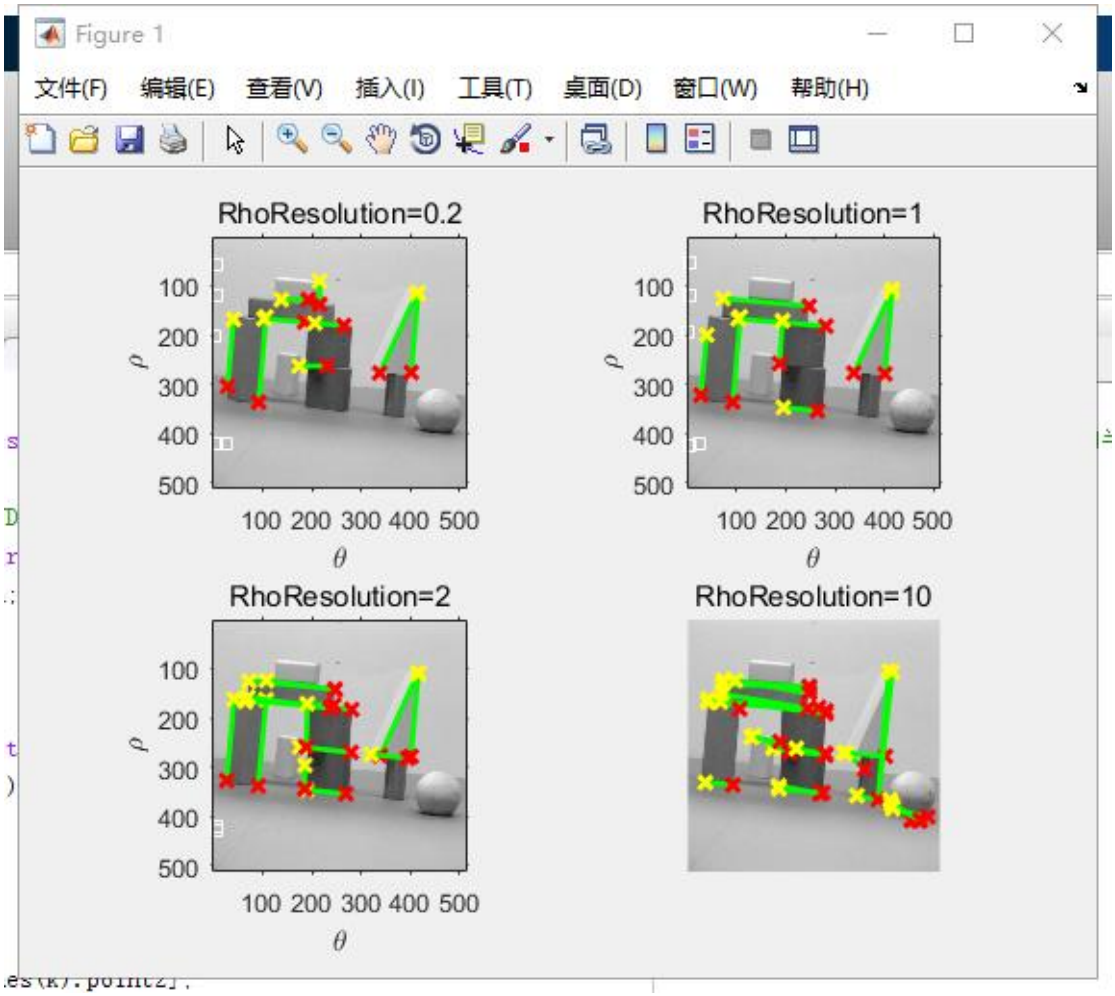
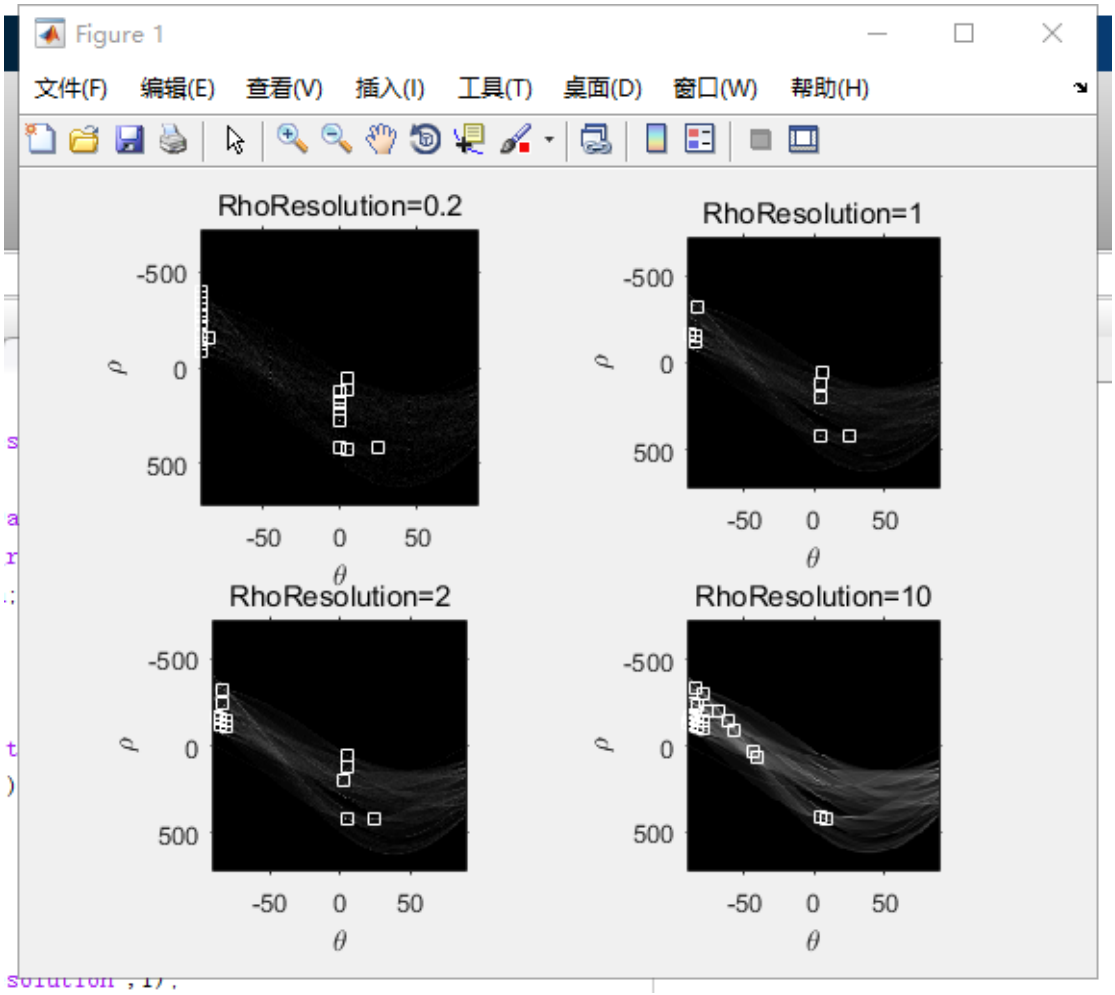


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
RhoResolution=0.2
RhoResolution=1
RhoResolution=2
RhoResolution=10
ρ
θ
100 200 300 400 500



结论：

因为 RhoResolution 指定在累计数组中（检测极值）的检测间隔默认为 1，RhoResolution 太大覆盖不到极值点，检测到一些不对应直线的次极值。由上方图像可以看出，当此参数设置为 0.2 时比设置为 1 与 2 时检测出的极值点更多。但是当此参数过大，比如为 10 时，却会将一些曲线当作直线处理，比如 test2 中的阴影部分。

改变 Theta 范围：

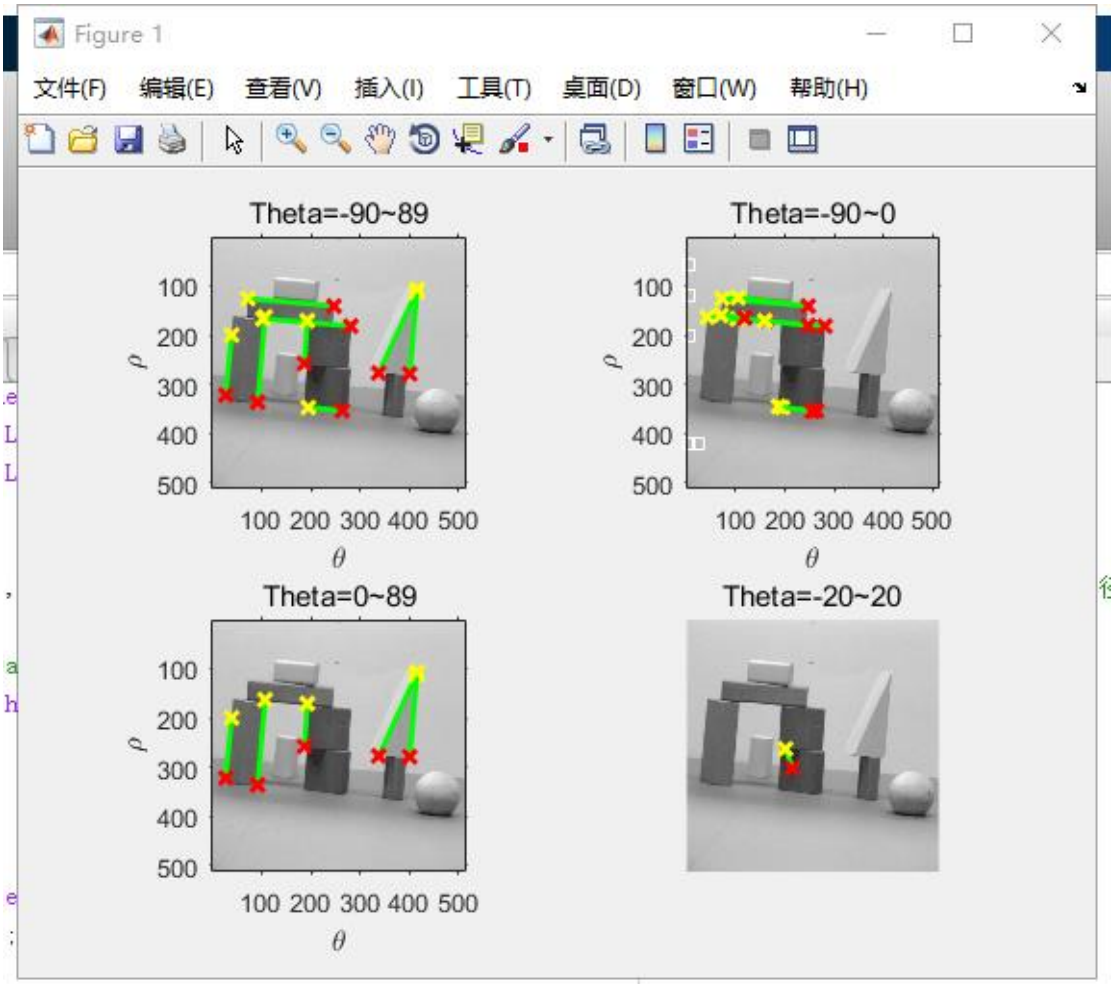
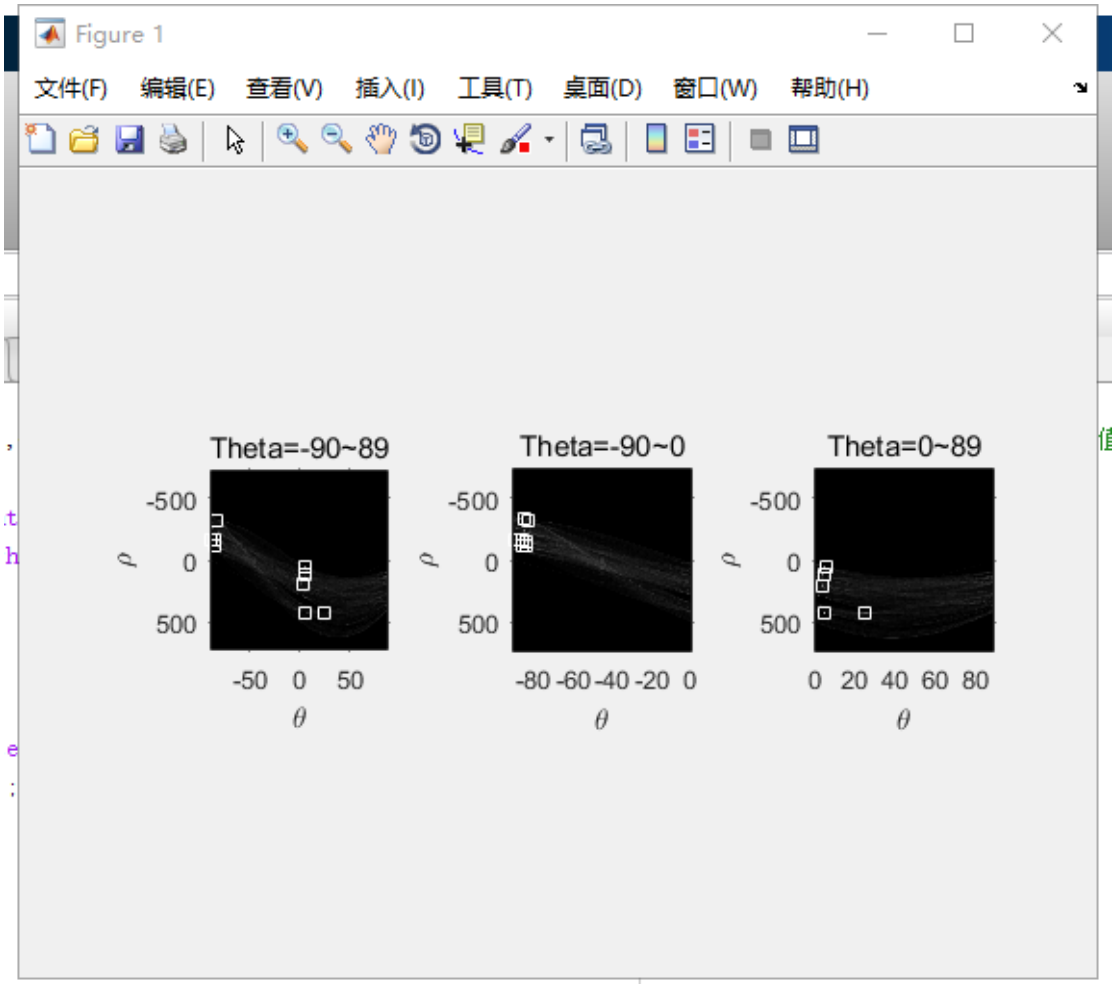


Figure 1
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
Theta=-90~89
Theta=-90~0
Theta=0~89
Theta=-20~20
100 200 300 400 500
ρ
θ



结论：

Theta 是指定检测的角度范围（不超过-90~90 度）以及间隔，例如-90:0.5:89.5，默认-90:1:89 Theta 范围设置小了显然会把某些极值排除在外，故 Theta 推荐使用全范围。如上图我们可以看到当 Theta 为全范围时，检测出所有的极值点。但是当 Theta 范围变为一半时（图 2 与图 3），检测出的极值点明显减少，对应检测出的直线数量也明显减少。

实验分析：

1. Sobel 算子是离散型的差分算子，用来计算梯度的近似值。该算子基于一阶导数而且引入了局部平均计算，有一定的消除噪声的功能。而且 Sobel 算子进行了加权处理，理论上来说比 Prewitt 和 Roberts 算子效果更好。但是 Sobel 算子没有将前景与背景完全区分。
2. Roberts 算子利用局部差分寻找边缘，检测垂直边缘效果较好，但是对噪声敏感，通常在图像边缘产生较宽的响应，所以边缘定位精度不高。
3. Prewitt 算子是一阶微分算子，能都去掉部分伪边缘，平滑噪声。

4. Canny 算子效果最好，但是复杂度较高。

5. 综上，我们如果只希望检测出图中的最明显的或者最主要的直线的时候，可以使用罗伯特，索贝尔或者 Prewitt 算子。但是如果我们希望检测出所有细微直线的时候，我们需要使用 LOG 或者 canny 算子。

6. Theta 是指定检测的角度范围（不超过-90~90 度）以及间隔，例如-90:0.5:89.5，默认-90:1:89 Theta 范围设置小了显然会把某些极值排除在外，故 Theta 推荐使用全范围。

因为 RhoResolution 指定在累计数组中（检测极值）的检测间隔默认为 1，RhoResolution 太大覆盖不到极值点，检测到一些不对应直线的次极值。