

# 数字图像处理

## 第七次作业：直线检测

自动化61 刘昊 2160504016

### 摘要

由于噪声、不均匀照明的影响，边缘检测得到的像素通常不能完全表征边缘，而是往往出现边缘断裂、杂散灰度不连续等现象。此时，常需要连接过程来组装像素成为有意义的边缘。本文首先分别利用两种边缘检测算法分别提取6张测试样片的边缘，再使用霍夫变换检测直线从而连接图像中线段。

## 1 边缘检测

### 1.1 实验原理

#### 1.1.1 Sobel边缘检测

Sobel算子是一种基于梯度信息进行边缘检测的算法。该算子中，定义两个方向上的梯度为：

$$g_x = \frac{\partial f}{\partial x} = (f(x+1, y-1) + f(x+1, y) + f(x+1, y+1)) - (f(x-1, y-1) + f(x-1, y) + f(x-1, y+1))$$

$$g_y = \frac{\partial f}{\partial y} = (f(x-1, y+1) + f(x, y+1) + f(x+1, y+1)) - (f(x-1, y-1) + f(x, y-1) + f(x+1, y-1))$$

梯度模板为

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

其中中心系数设为2是为了更好的平滑图像抑制噪声，若设为1则为Prewitt模板。

梯度幅值为：

$$M(x, y) = g_x^2 + g_y^2$$

实际中为了简化计算，可以牺牲各向同性将上式近似为

$$M(x, y) \approx |g_x| + |g_y|$$

经上述计算后，某些像素的灰度值可能超出设定范围，此时注意需要重新标定。

#### 1.1.2 Canny算法

Canny边缘检测算法于1986年由John Canny首次提出，它是从不同视觉对象中提取有用结构信息并大大减少要处理的数据量的一种技术，目前广泛应用于各种计算机视觉系统。

Canny算法基于三个基本目标：

1. 低错误率检测到所有边缘，没有伪响应；
2. 边缘点尽量被定位在真实边缘的中心；
3. 单一的边缘点响应。

算法基本处理流程为：

1. 使用高斯滤波器平滑输入图像，滤除噪声，滤波器模板一般选择 $6\sigma \times 6\sigma$ 大小；
2. 计算梯度幅值图像和角度图像，其中：

$$M(x, y) = \sqrt{g_x^2 + g_y^2}$$

$$\alpha(x, y) = \arctan\left[\frac{g_y}{g_x}\right]$$

3. 对梯度幅值图像应用非最大抑制，即：

① 将当前像素的梯度强度与沿正负梯度方向上的两个像素进行比较。梯度方向像素点可以选择8邻域中方向最接近梯度方向的像素点，或是采用8邻域中在该方向两侧的相邻像素点线性加权求得，例如：假设某个像素点的梯度方向 $\alpha$ 在 $[0^\circ, 45^\circ]$ 内，则该像素梯度方向上两像素幅值为

$$M_{p1} = (1 - \tan(\alpha)) \times M(x + 1, y) + \tan(\alpha) \times M(x + 1, y + 1)$$

$$M_{p2} = (1 - \tan(\alpha)) \times M(x - 1, y) + \tan(\alpha) \times M(x - 1, y - 1)$$

② 如果当前像素梯度强度均大于另外两个像素，则该点保留为真实边缘点，否则被抑制，置0。

4. 双阈值处理与连接分析

① 设定双阈值，所有大于高阈值的像素点视作强边缘像素；所有高于低阈值低于高阈值的像素点暂时标记为弱边缘像素；其他像素不做标记。

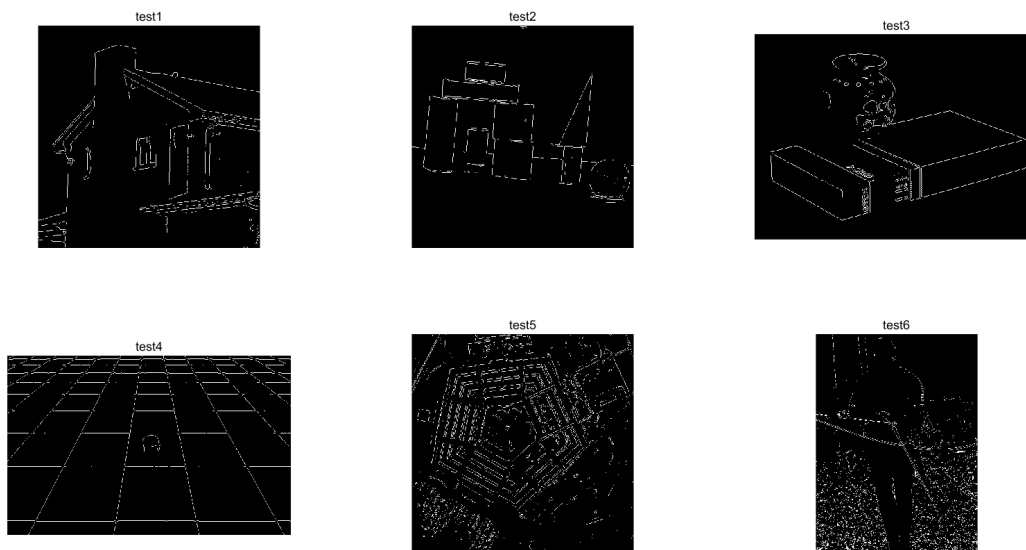
② 对于弱边缘像素，为了跟踪边缘连接，检测其8邻域的像素灰度值，若存在强边缘像素，则认为该若边缘像素连接了强边缘，属于真实边缘，标记为有效边缘像素，否则删除标记。

③ 将所有非有效边缘像素和强边缘像素的像素点灰度置0，其他点保留或置1。

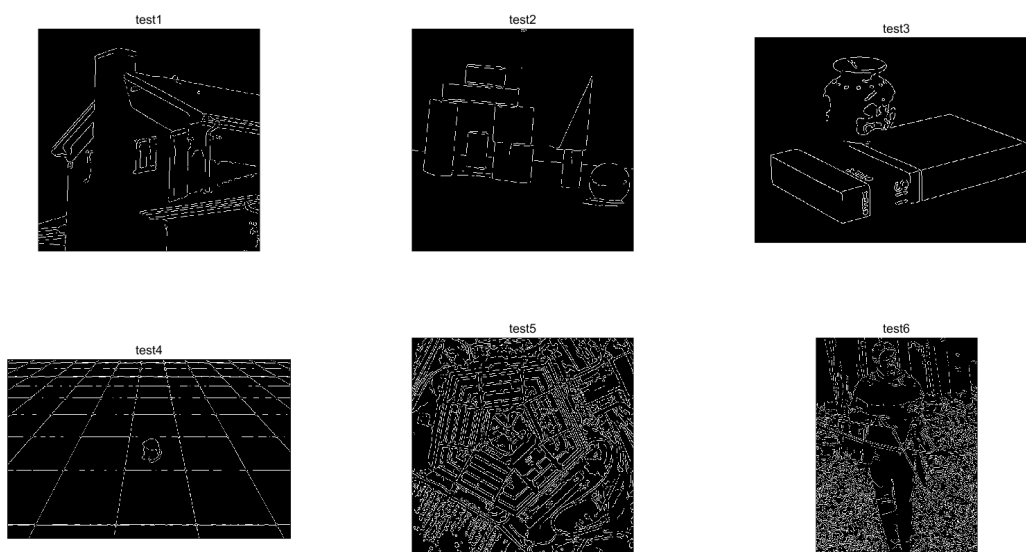
## 1.2 实验结果

利用Matlab软件编程，使用两种边缘检测算法分别提取6张测试样片的边缘，结果如下：

Sobel边缘检测结果：



Canny边缘检测结果：



注意，图像中线条视觉上的“断裂”，多数只是因为粗细不均等或者锯齿形线条缩放引起的，实际上是连接的线条。

## 2 直线检测

### 2.1 实验原理

霍夫变换（Hough Transform）是一种识别直线的方法，其基本原理是利用点与线的对偶性，将原始图像控件的给定直线通过曲线表达形式转变为参数空间的一个点。这样就把原始图像中给定直线的检测问题转换为寻找参数空间中的封之问题。

其直线检测算法流程为：

1. 使用边缘检测算法将输入图像转换为二值化边缘图像；

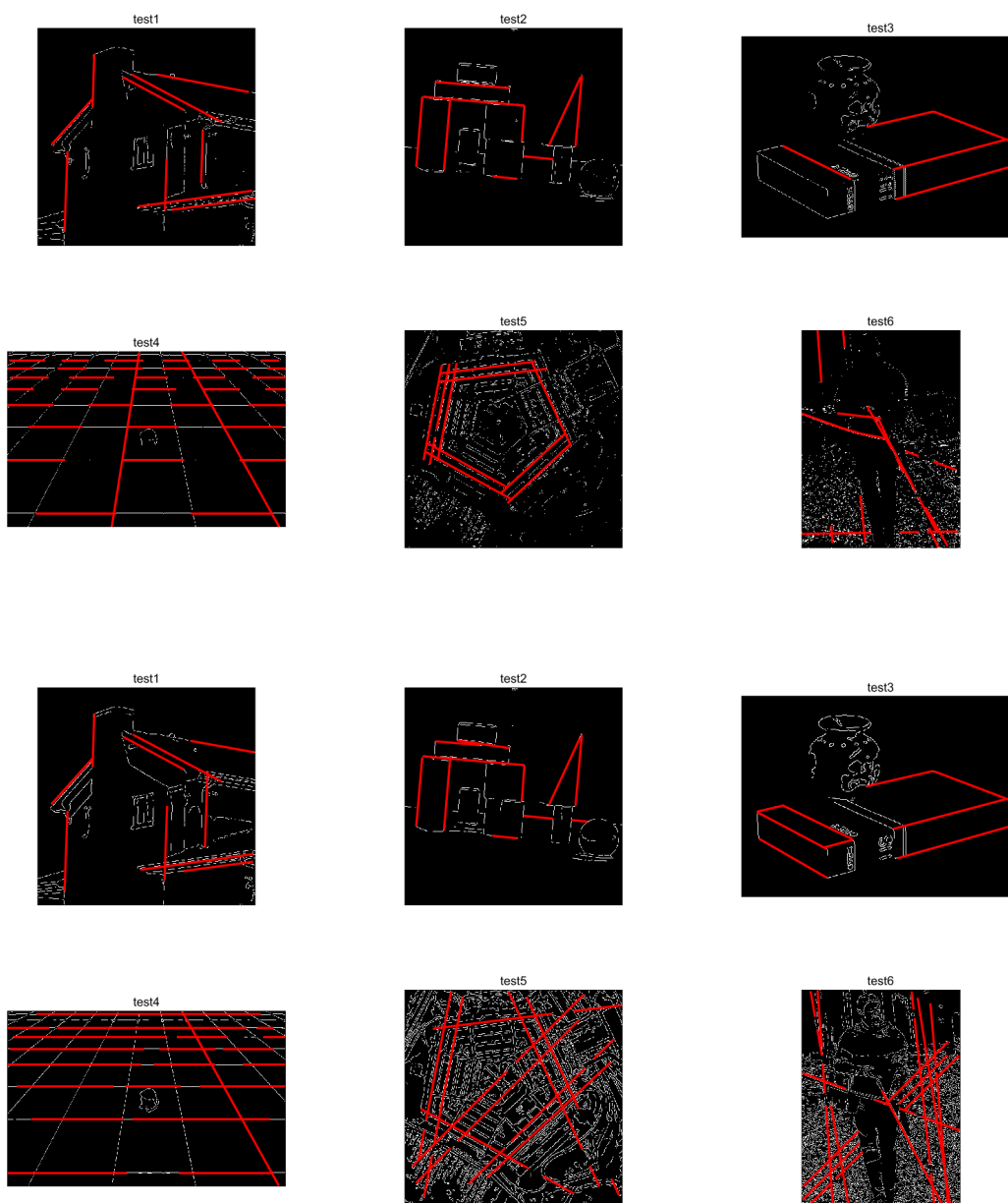
2. 对该图进行霍夫变换，即指定 $\rho\theta$ 平面中的细分，统计各累加单元的值，得到霍夫变换矩阵；
3. 使用峰值检测函数找到大于阈值的霍夫变换单元；
4. 将上述识别出的一组候选峰，确定与其相关的线段及其起止点；
5. 描绘于结果图上。

其中， $\rho\theta$ 平面两轴的细分值会对直线检测结果产生较大影响。

## 2.2 实验结果

### 2.2.1 不同算法对直线检测的影响

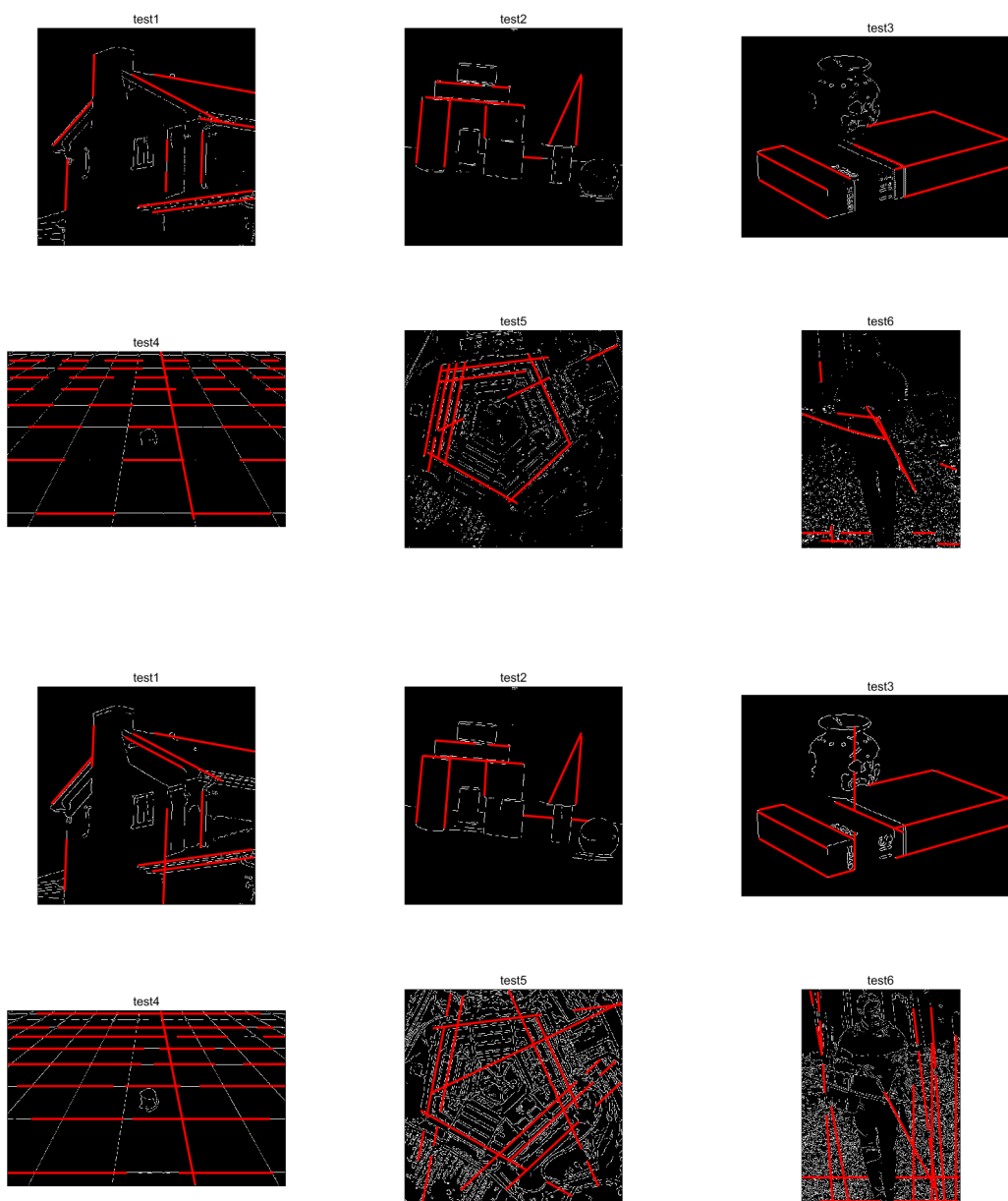
首先令 $\rho\theta$ 两轴细分值均为1，阈值默认为0.5倍最大值，检测直线数量最多为10，Sobel边缘检测与Canny算法所得二值图像的检测结果分别为：



可见，对于边缘细节较粗糙、形状较简单的二值图像，霍夫变换能够较为准确的检测出图像中的直线，而对于纹理较复杂的图像，霍夫变换则容易被纹理“误导”，产生伪峰，即检测出图像中不存在的直线。因此，对于边缘较为简单、纹理较少的图像，使用Canny算法提取出的边缘更精细，故此时霍夫变换效果较好（如test3图），但对于纹理复杂的图像，Canny变换往往无法忽略一些图像细节，反倒降低了直线检测效果，此时增大双阈值差距或者采用Sobel算子效果较好（如test5图）。

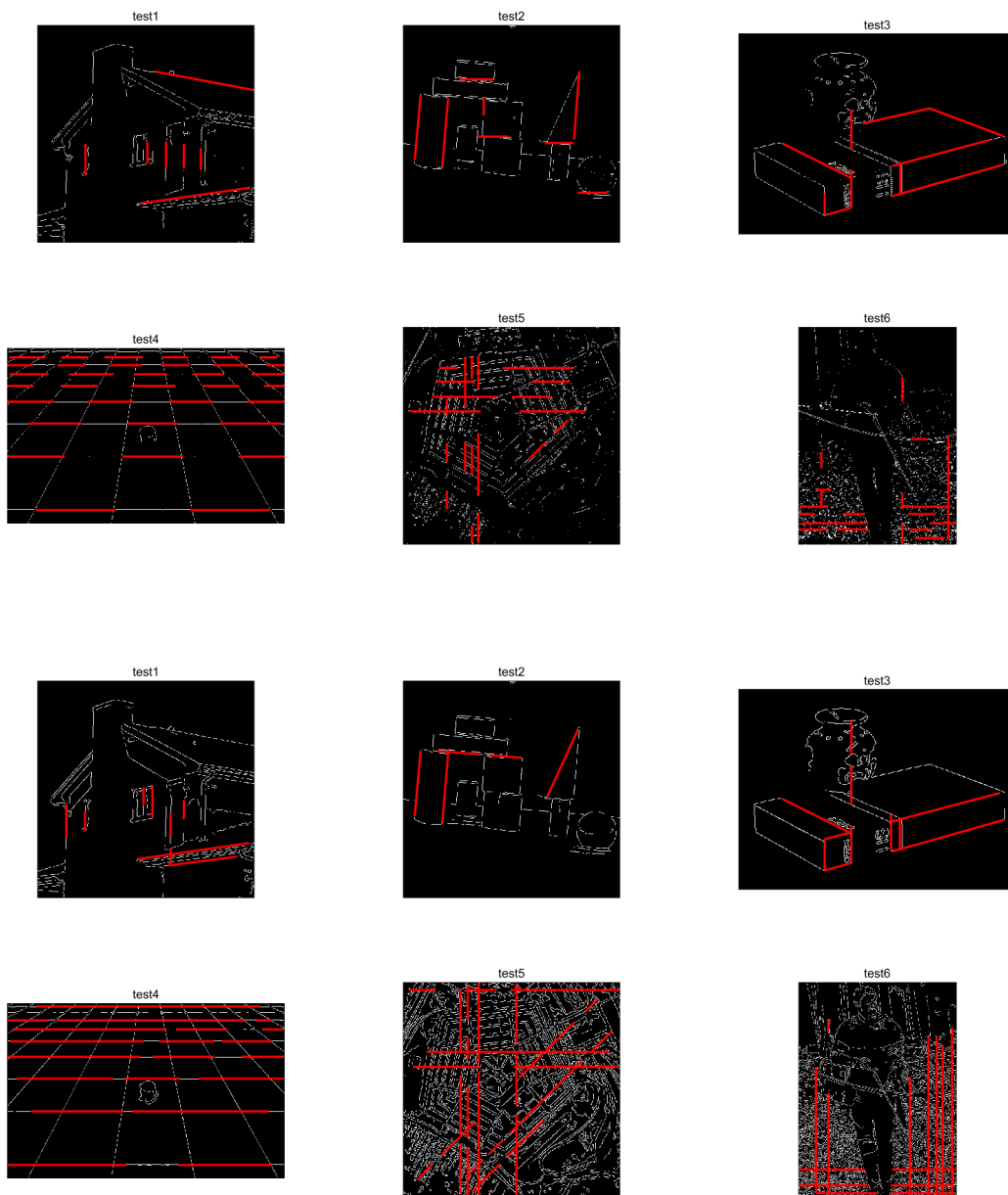
### 2.2.2 不同 $\rho\theta$ 细分值对直线检测的影响

适当减小 $\rho\theta$ 细分值至0.8，结果如下：



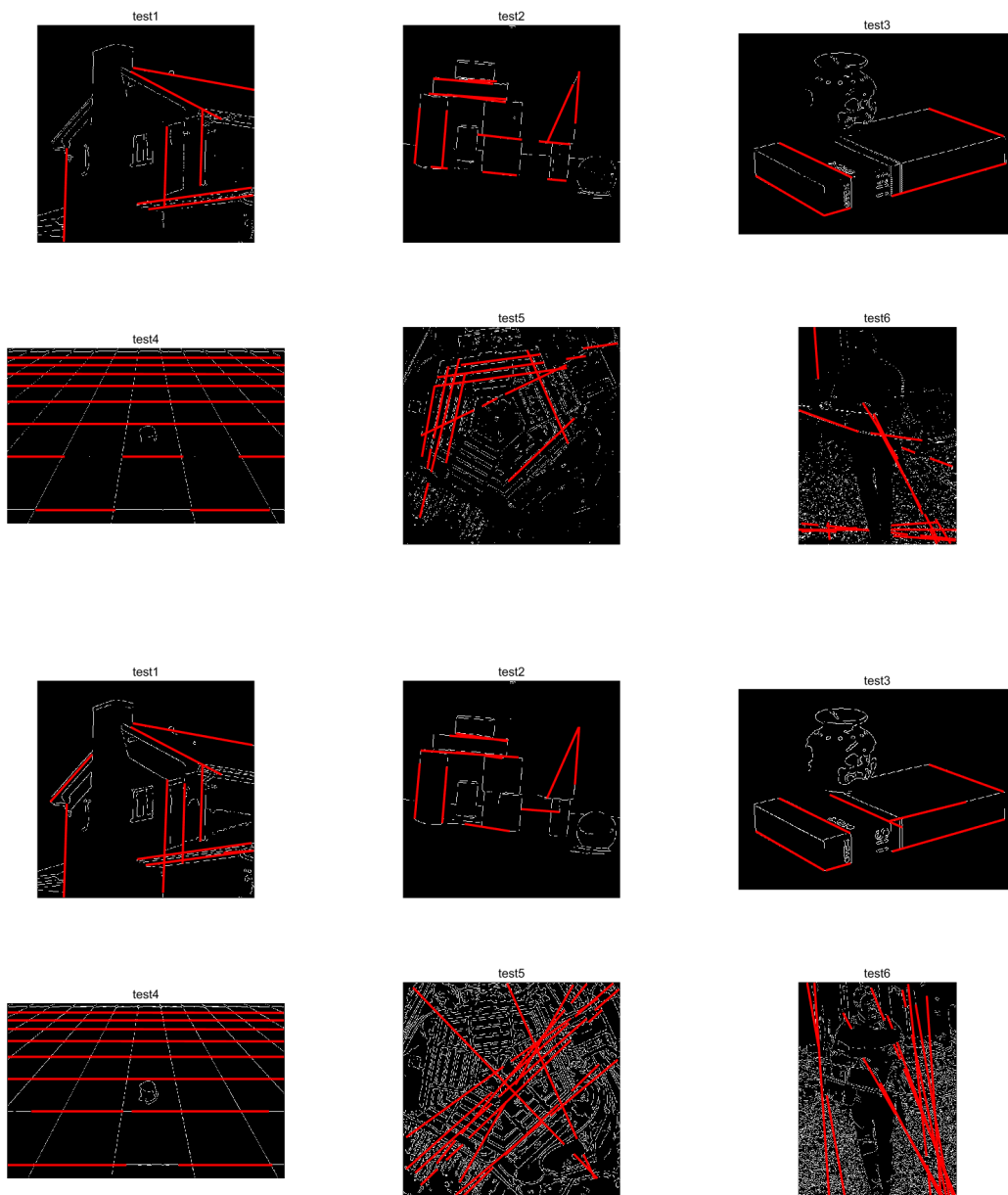
可见，减小 $\rho\theta$ 细分值有助于提高直线检测精度，如Sobel处理的test3和Canny处理的test5，直线检测效果都有明显提升，但注意Sobel组的test2和Canny组的test3，细分精度上升也使得较粗糙的边缘违背检测到，以及也造成的误判直线的出现。

继续减小细分值至0.2，结果如下：



此时误判现象十分严重，可见细分精度并非越大越好。

下面我们考虑增大细分值至2，结果如下：



可见，当细分值较大时，细分精度较低，直线检测精度也更低，受细节纹理影响更严重。但同时，细分精度降低也带来的计算复杂度较低的优势。

综合以上分析可知， $\rho\theta$ 细分值并非越小越好，过大的细分值带来较低计算复杂度同时也带来较低的检测精度，过小的细分值虽然检测精度较高，但容易造成直线的误判，即出现“伪峰”。因此实际操作时需要根据不同的图像选用合适的细分值。