数字图像处理 - 直线检测

自动化钱 61　柴嘉骏 2160405071

2019 年 5 月 16 日

目录

1 边缘检测

边缘检测是基于灰度突变来分割图像的常用方法，一阶导数的幅度可用于检测图像中的某个点处是否存在一个边缘，而二阶导数的符号可以用于确定一个边缘像素是位于暗侧还是亮侧，但是在本次实验中我们并不需要二阶导数信息。作为基本边缘检测的几种方法，我们使用梯度信息来表征与识别图像的边缘，使用梯度信息的常用算子为 Sobel 算子，其基于 x 方向和 y 方向的梯度信息检测边缘，以 3*3 的模板为例，其偏导数由下式给出：

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3) \tag{1}$$

和

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

具体的算子如下所示：

$$Sobel_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, Sobel_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{3}$$

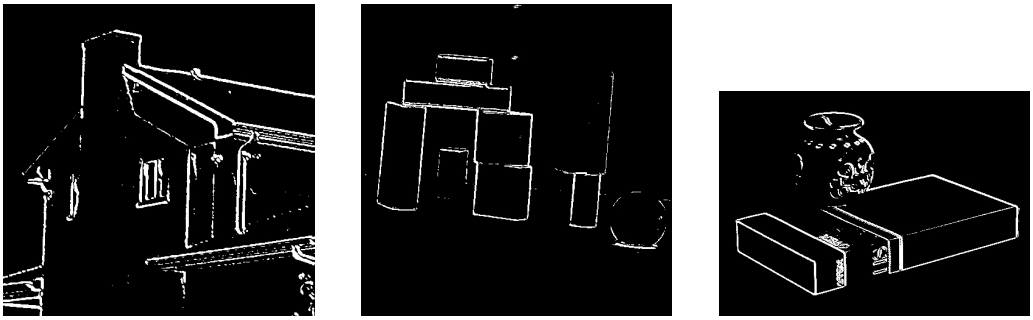
除此之外，还有一些更先进的边缘检测技术，其以一个模板或多个模板对图像进行滤波为基础，而未对图像特性和噪声内容采取预防措施，例如 Marr-Hildreth 边缘检测器，其使用诸如 Sobel 算子等较小的模板寻找图像的零交叉；Canny 边缘检测器则基于低错误率、更好定位边缘点和单一边缘点响应的基本目标进行边缘检测。在本次实验中使用 Canny 边缘检测器作为先进方法的展示。

1.1 Sobel 边缘检测器

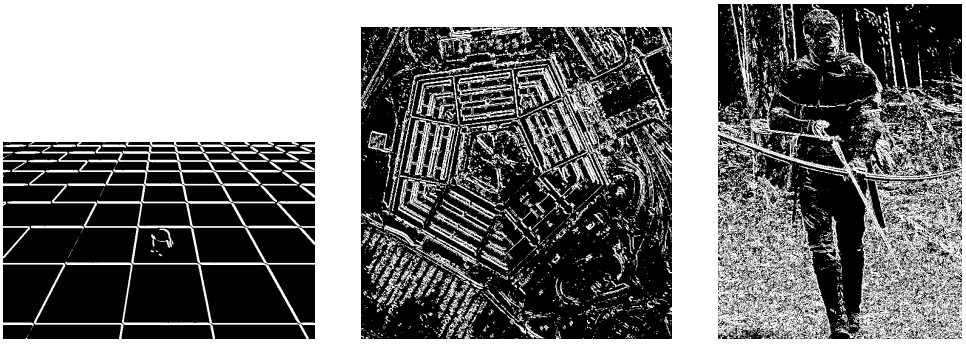
对六个测试图像使用 Sobel 算子进行边缘检测得到的结果如图 1所示：

$$Sobel_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}, Sobel_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{4}$$

由上文中介绍的 Sobel 算子可以看到，使用 Sobel 算子进行边缘检测应为 x，y 两个方向上的，但是在本次实验中为了后续实验中的直线检测更好进行，我将 x 方向与 y 方向上进行 Sobel 边缘检测结果进行相加，得到了图 1所示的结果，可以看到该边缘检测的效果还是比较可观的，能够实现在 x 和 y 两个方向上的边缘检测，但是可以很明显的看到在倾斜方向上的边缘检测效果并不是很好，可以使用不同的算子进行改进但并不在本次实验的计划之中。



(a) $Size = 3pixels$ (b) $Size = 5pixels$ (c) $Size = 7pixels$

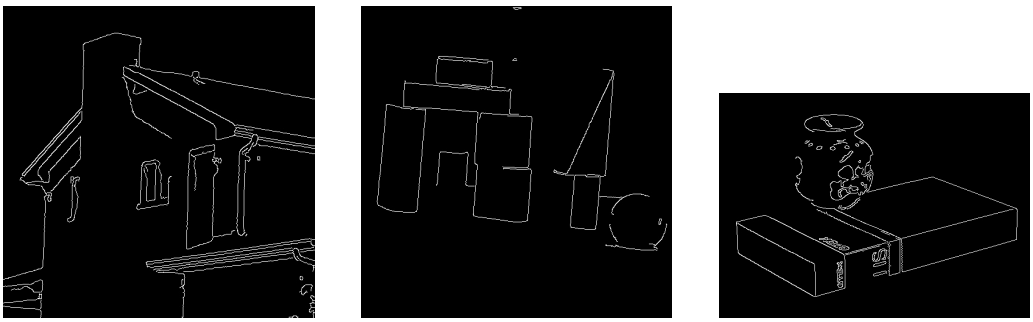


(d) $Size = 3pixels$ (e) $Size = 5pixels$ (f) $Size = 7pixels$

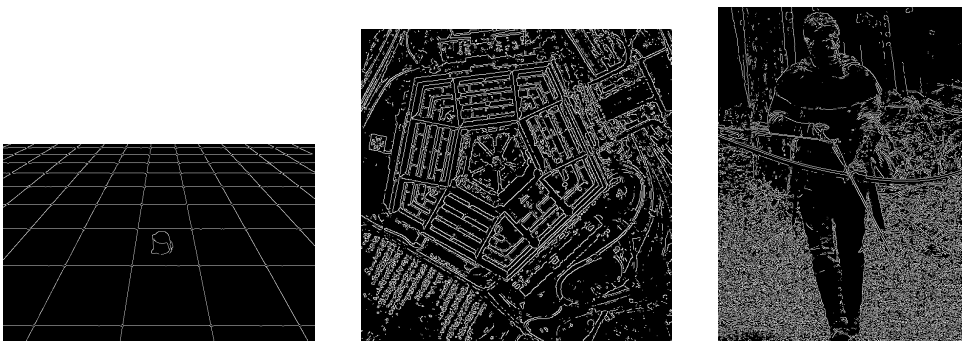
图 1: 对测试图像进行 Sobel 边缘检测

1.2 Canny 边缘检测器

本次实验使用的模板为如下所示的 x 方向和 y 方向的 Sobel 算子, 对六个测试图像使用 Canny 边缘检测器进行边缘检测得到的结果如图 2所示:



(a) $Size = 3pixels$ (b) $Size = 5pixels$ (c) $Size = 7pixels$



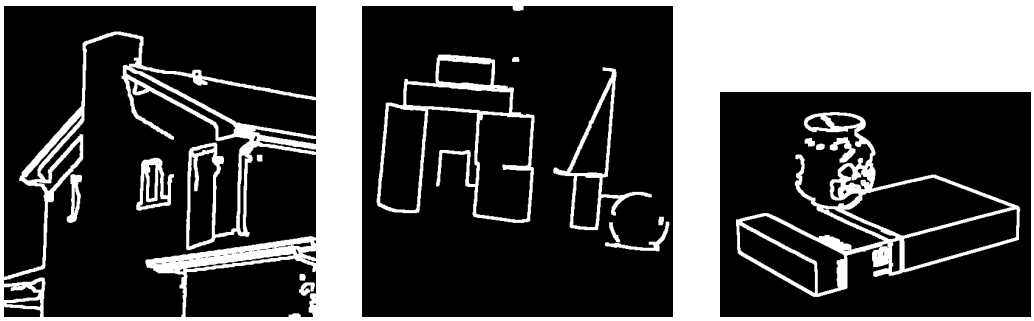
(d) $Size = 3pixels$ (e) $Size = 5pixels$ (f) $Size = 7pixels$

图 2: 对测试图像进行 Canny 边缘检测

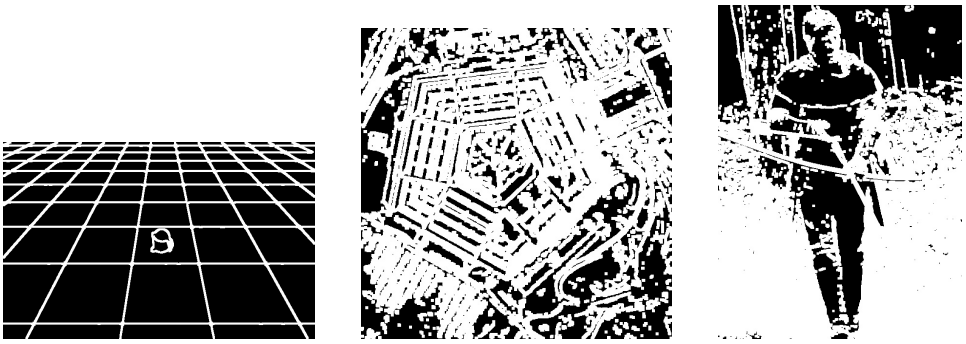
Canny 边缘检测的过程是比较复杂的，在教科书中有详细的介绍就不在此赘述了，不过我们可以通过与 Sobel 边缘检测的结果进行对比看到，在 Canny 边缘检测中不仅斜边的边缘被轻易地检测出来，检测出的边缘更是十分的清晰与连续，这与 Canny 方法的复杂性是密不可分的。

1.3 膨胀 Canny 算子

在实验的过程中，由于 Canny 方法得到的边缘较细，其直线信息不能够完整地保留，甚至在某些肉眼可辩的连续边缘处存在中断，因此对 Canny 边缘检测得到的结果进行膨胀操作，得到了如图 3的结果：



(a) $Size = 3pixels$ (b) $Size = 5pixels$ (c) $Size = 7pixels$



(d) $Size = 3pixels$ (e) $Size = 5pixels$ (f) $Size = 7pixels$

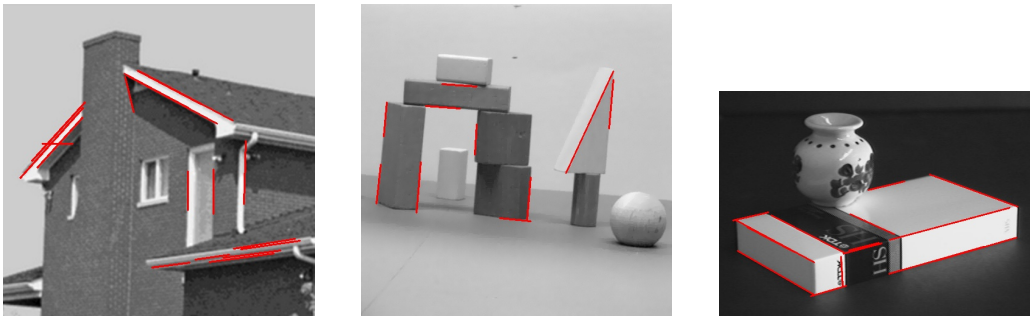
图 3: 对测试图像进行 Canny 边缘检测

2 Hough 变换直线检测

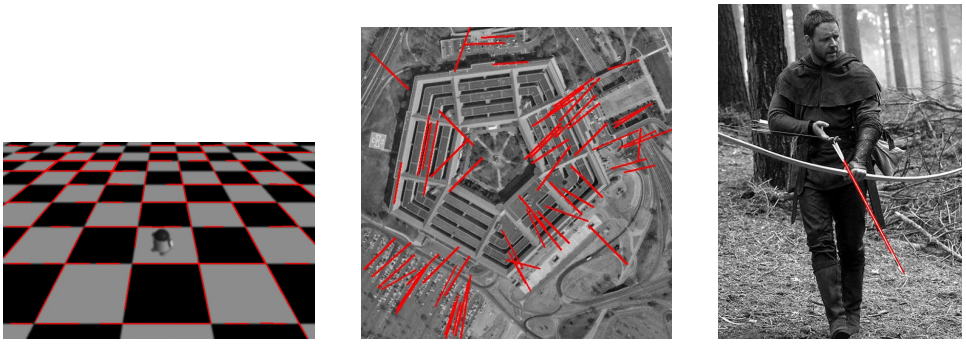
一条线可以表示成 $y = mx + c$ 或参数形式，例如 $\rho = xcos\theta + ysin\theta$，其中 ρ 是从原点到直线的垂直距离，θ 角是由这条垂线和水平轴以逆时针的方向形成的。同上，Hough 变换在教材中已经介绍了很多，因此再本报告中不进行赘述。

2.1 对 Canny 检测结果进行 Hough 变换

对使用 Canny 检测得到的边缘信息进行霍夫变换检测图中的直线，对六个测试图像进行测试得到的结果如图 4所示：



(a) $Size = 3pixels$ (b) $Size = 5pixels$ (c) $Size = 7pixels$



(d) $Size = 3pixels$ (e) $Size = 5pixels$ (f) $Size = 7pixels$

图 4: 对测试图像进行 Canny 边缘检测

从图中可以看到直线检测的结果并不是很好，更尤其是这六个测试图像的结果并不是完全自动生成的，而是针对每一个测试图像手动选择合适的算法参数才能得到如此结果。在 OpenCv 中存在至少两种 Hough 变换的方法，本次展示的是 HoughLinesP 得到的结果，在 HoughLines 函数得到的结果中，第五、六幅测试图像甚至出现了全红的现象，其原因可以从第一部分的边缘检测中看到，其存在无数的边缘信息，在进行 Hough 变换时将会被距离十分紧密的虚假边缘信息欺骗，使算法误以为这是一个断了的很长的边缘信息，为此在这些方法中提供了 maxLinesGap 参数用于区别直线间的距离。

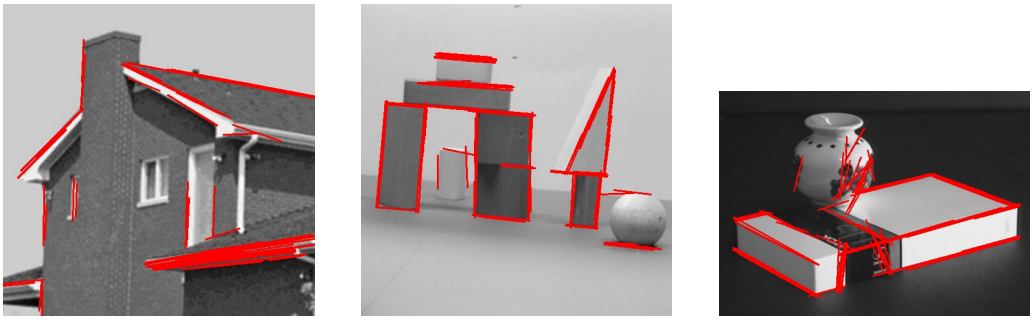
为了得到更多的直线信息而不致产生混淆，我使用了进行膨胀后的边缘图像进行直线检测，得到了如 tu 图 5所示的结果：

可以看到除了第五、六幅测试图像外，在 Hough 变换参数保持一致的前提下，其他图像的边缘信息都得到了很好的增强。

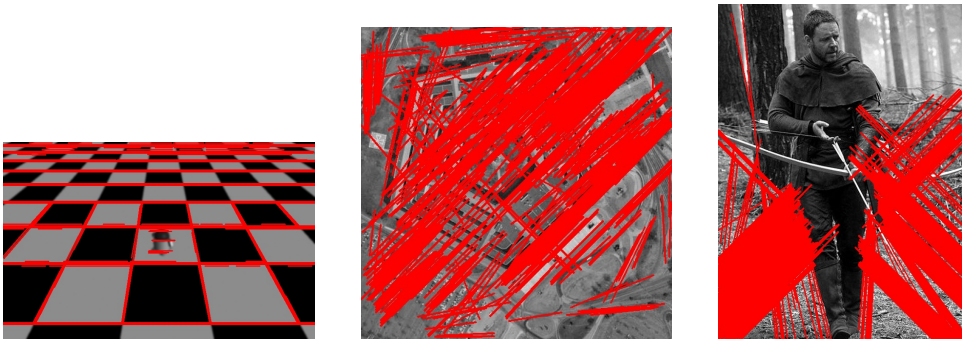
2.2 对 Sobel 检测结果进行 Hough 变换

使用 Sobel 检测得到的边缘信息进行霍夫变换检测图中的直线，对六个测试图像进行测试得到的结果如图 6所示：

可以看到，在 Hough 变换参数相同的情况下，使用 Sobel 检测得到的边缘信息将提供更多的直线结果，但是这些直线结果是完全冗余且虚假的，究其原因在于 Sobel 检测方法提取出的边缘信息受噪声因素影响太大，和前面叙述的原因类似，这些噪音将会对算法构成干扰，让其误以为

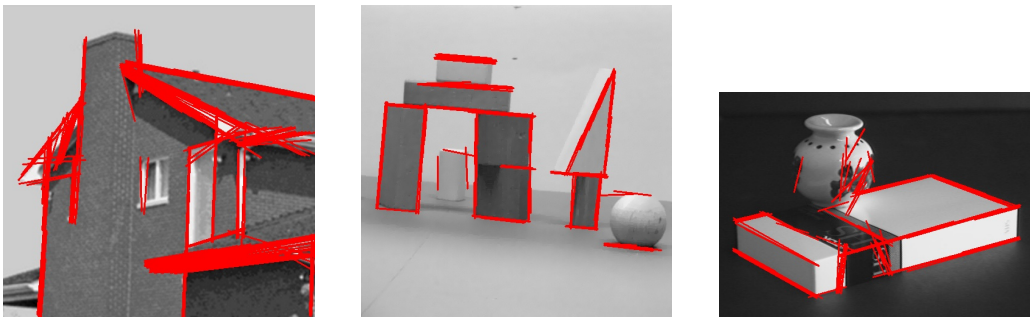


(a) $Size = 3pixels$ (b) $Size = 5pixels$ (c) $Size = 7pixels$

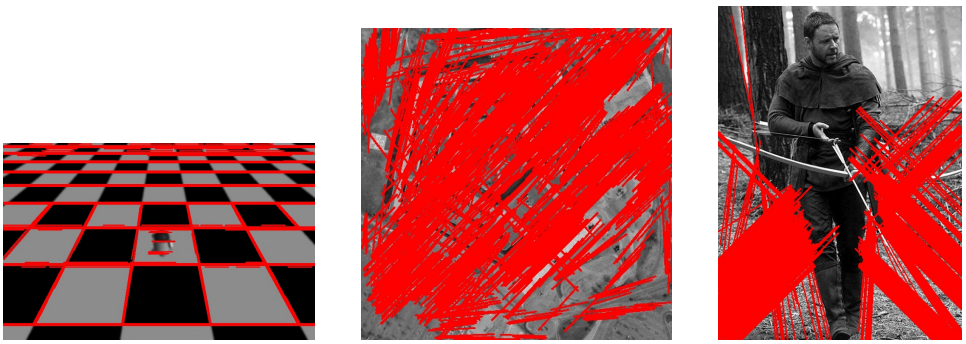


(d) $Size = 3pixels$ (e) $Size = 5pixels$ (f) $Size = 7pixels$

图 5: 对测试图像进行 Canny 边缘检测



(a) $Size = 3pixels$ (b) $Size = 5pixels$ (c) $Size = 7pixels$



(d) $Size = 3pixels$ (e) $Size = 5pixels$ (f) $Size = 7pixels$

图 6: 对测试图像进行 Canny 边缘检测

是断了的长直线导致错误识别率大大提升。

2.3 Hough 变换参数的影响

上述实验均为手动调试参数得到的较为不错的结果，在 OpenCv 中的 HoughLinesP 函数里关键的参数有：rho, theta, threshold, lines, minLineLength 和 maxLineGap。其中：

image 参数表示边缘检测的输出图像，该图像为单通道 8 位二进制图像。

rho 参数表示参数极径以像素值为单位的分辨率，一般使用 1 像素。

theta 参数表示参数极角以弧度为单位的分辨率，一般使用 1 度。

threshold 参数表示检测一条直线所需最少的曲线交点。

lines 参数表示储存着检测到的直线的参数对的容器，也就是线段两个端点的坐标。

minLineLength 参数表示能组成一条直线的最少点的数量，点数量不足的直线将被抛弃。

maxLineGap 参数表示能被认为在一条直线上的亮点的最大距离。

增加 minLineLength 可以减少直线个数，增加 maxLineGap 可以增加直线个数，具体的实验由于参数与测试图像过于庞大难以在此展示，如对我的完成情况有疑问可以运行我附件中的程序。

3 总结

本次实验实现了基于 Sobel 算子和 Canny 检测的边缘检测器并调整参数得到了较好的结果，又使用 Hough 变换对 Sobel 算子、Canny 方法得到的边缘信息以及膨胀后的 Canny 方法的边缘信息进行 Hough 变换直线检测，发现了 Hough 变换参数对于检测结果的巨大影响，并通过手动调参获得了较好的检测结果，同时在参数相同的前提下对不同边缘信息进行直线检测的结果进行比较并分析原因。

本次实验的收获还是比较大的，也让我加深了对 Canny 边缘检测和 Hough 变换直线检测有了更深的理解。

注：本次实验使用的平台为 Python2.7 + OpenCv。