

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

2016-4-29

HOMEWORK7

Several thin, curved lines in dark blue and light gray originate from the bottom left corner and curve upwards and to the right.

Cecil Wang
PEKING UNIVERSITY

目录

一、作业描述与算法设计	2
二、实现细节	2
2.1 Canny 边缘检测	2
2.2 Hough 变换	3
三、实验结果	4

一、作业描述与算法设计

作业描述：实现一个在图中标出所有圆形的算法。

算法设计：算法分为两步。首先对图片进行灰度处理，之后使用边缘检测算法标出图片中的所有边缘；然后在边缘图中使用 Hough 变换找出图中所有可能为圆形的边缘。

二、实现细节

2.1 Canny 边缘检测

在边缘检测的算法中我选择了 Canny 边缘检测，原因之一是 Canny 算法检测的边缘更准确，其次是因为 Canny 作为课上的补充，在讲授的时候并没涉及太多。

Canny 边缘检测算子是 John F. Canny 于 1986 年开发出来的一个多级边缘检测算法。其流程大致可以分为以下 4 个步骤：

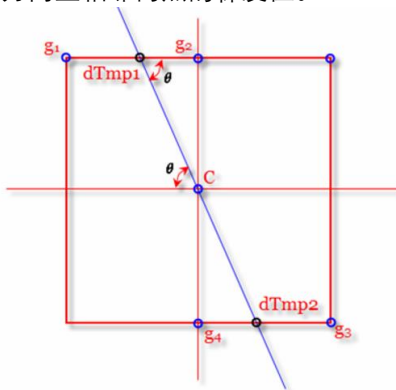
1. 应用高斯滤波平滑图像去噪声
2. 使用边缘检测算子计算各个像素点的梯度
3. 使用非最大抑制消除提取局部最大梯度的点
4. 使用双阈值方法从强梯度点扩展边缘

在我的实现中，为了方便我的边缘检测算子选择了 Sobel 算子，而且只检测了水平方向和竖直方向的梯度。

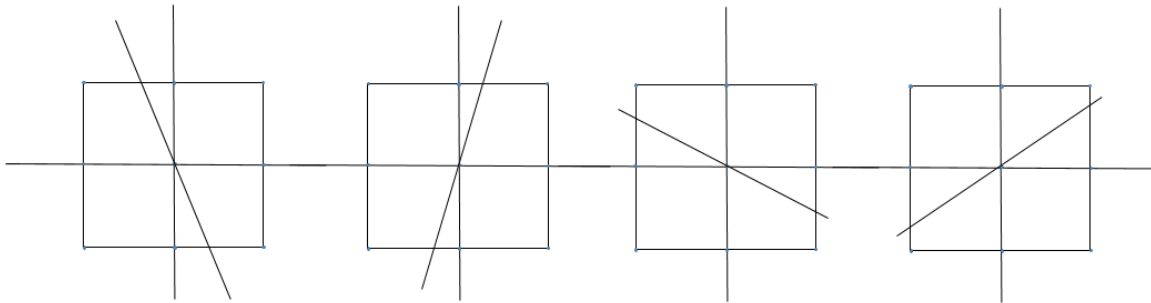
$$s_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, s_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad K = \begin{bmatrix} a_0 & a_1 & a_2 \\ a_7 & [i, j] & a_3 \\ a_6 & a_5 & a_4 \end{bmatrix}$$

步骤1与步骤2非常简单，只需要用算子与每个像素点及其邻域做卷积即可，我直接通过 matlab 的内置函数 `imfilter`。下面将详述非最大抑制与双阈值方法。

非最大抑制实际上是选取局部上最大值的算法，在 Canny 算法中，一个像素点是局部梯度最大值被定义其梯度值大于其梯度方向上相邻两点的梯度值。



如上图所示，c为当前的像素点，蓝色直线为c的梯度方向，而c的相邻点为dTmp1与dTmp2。由于dTmp1与dTmp2可能不是真正存在的像素点，所以使用插值法通过g1、g2计算dTmp1的值，通过g3、g4计算dTmp2的值。如果c的梯度小于dTmp1与dTmp2的梯度值则将c的梯度值设为0，否则不变。如果对上图稍作变化，我们可以得到全部的四种情况：



对四种情况做出相应的插值计算即可。

双阈值法我的理解实际上是一个DFS算法，首先枚举图片中的每一个像素点，如果它的梯度超过强阈值则将它加入到边缘点集合同时作为一次DFS的起点，搜索它的八连通邻域内的像素点，若有像素点的梯度超过弱阈值则也将它加入到边缘点集合同时对它也做同样的DFS。

总结Canny边缘检测的算法精华所在是只相信那些梯度大的点为边缘点，同时通过这些点不断地扩展其他的边缘点，在扩展的过程中只有梯度大于弱阈值的点才可以通过检测。这样的方法可以有效地减少噪声与干扰的影响。

2.2 Hough 变换

霍夫变换(Hough Transform) 霍夫变换是图像处理中从图像中识别几何形状的基本方法之一。其主要思想是通过几何图形的数学表达式建立一个参数空间，例如对于圆又如下公式：

$$x = x_0 + r \cos \theta$$

$$y = y_0 + r \sin \theta$$

其中 x_0 为圆心的横坐标， y_0 为圆心的纵坐标， r 为圆的半径， θ 表示圆上点与圆心的夹角，取值范围为 $[0, 2\pi]$ 。那么 Hough 空间即为 (X, Y, R) ，通过特定取值的 x_0 ， y_0 ， r 可以唯一确定一个圆。对于图中一个点，我们将 Hough 空间中所有包含此点的圆对应的点进行计数。在最终的 Hough 空间中，取值大的点在概率意义上最有可能为图中存在的圆。

虽然 Hough 算法思想简单，实现容易，但是应对简单情况效果非常好，不过对于 Hough 算法参数的选取确定是一个非常考验技巧的事情，具体的讲 Hough 算法参数包括 Hough 空间 R 域的确定、 θ 取值的粒度、最终认为可信圆对应的的点的值的下限和最终反映射到图中点时半径的模糊范围。

Hough 空间 R 域的范围直接决定了最终结果图中找出的圆的大小。

θ 取值的粒度，如果粒度太细会加大噪声的干扰，但是粒度太粗又会降低“浅圆”（即那些圆边界上点的密度低）被检测出来的可能性。

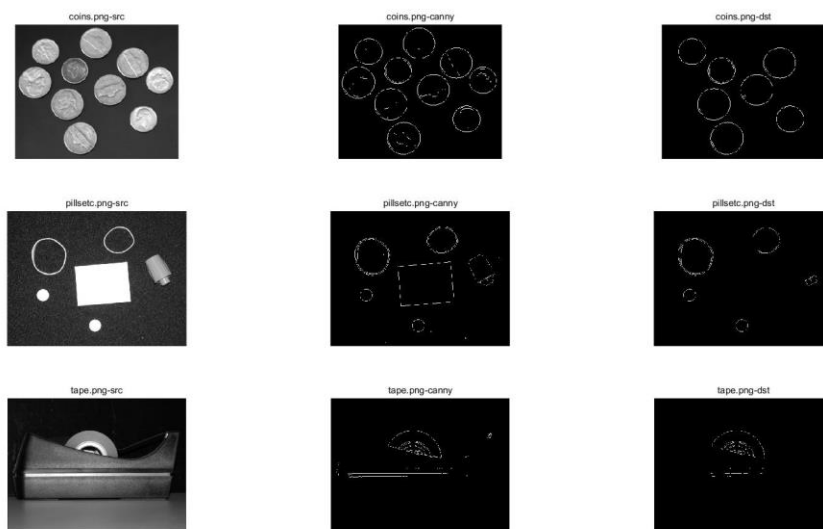
同时以上两者决定了算法的时间复杂度。

最终认为可信圆对应的的点的值的下限直接影响了噪声在结果图中的所占比，但是当不规则圆较多时，这个下限需要降低，而当规则园较多时应该提高下限。

最终反映到图中点时半径的模糊范围决定了最终显示的点的多少，当范围小时圆可能不清晰，但是范围大又会包括进去噪声。

对于不用的图应该综合考虑各种因素，才能准确的检测出圆形，这也意味着一个好的 Hough 变换实现要提供及其细致的参数设置接口。另外，Hough 变换可以加入边缘点梯度的剪枝，例如当寻找圆是，如果结果点集中存在一个子集且子集中点在局部上靠近且梯度一直，则它们表示的为一条直线，可以直接排除不是圆形。

三、实验结果



在此对实验结果进行一些补充说明：

- 1.对于 coins.png 图片由于硬币的边界非常光滑且在 canny 处理之后，噪声边缘较少，可见最终的结果很好，此时 hough 算法的阈值设定为 0.85
- 2.对于 pillsetc.png 图片可见图片上部两个圆形非常不规范，为了最终结果显示出这两个圆，将 hough 算法的阈值设定为 0.4
- 3.对于 tape.png 图片其难点在于图片中部存在一个长条的干扰，若要去除这个干扰其实也很容易，我们可以将 hough 空间的圆半径范围的下界提高。这样操作可行的原因是当半径过小的时候长条带可以看做是若干个水平排列的圆的顶部端点的切线与底部端点的切线的组合而成，所以最终结果会将其显示。



上图是将半径范围提高后的结果，虽然仍然存在一些残余的干扰但是较以前相比已经很多，而之所以仍存在残余是因为残余的点会被拟合到图片显示的园中。