

计算机视觉第三次作业

朱明杰 15331441

一. 实验任务

1. 输入普通 A4 打印纸，上面可能有手写笔记或者打印内容，但是拍照时可能角度不给出正。输出图像的边缘、A4 纸边缘的各直线方程以及 A4 纸的四个角点。
2. 输入一张照片，其中包含若干硬币，输出图像的边缘、把图像中边缘拟合成圆以及输出图像中硬币的数量。

二. 实验工具

Visual studio 2015

Clmg Library

三. 算法流程

1. Canny 算子。

直接利用上次作业所实现的 Canny 算子代码即可。不同的是，这次 Canny 算子要把阈值卡得非常高，因为有一张图是里面也有条贯穿 A4 纸的直线的。如果不把阈值卡死的话，贯穿 A4 纸的那条直线就会对之后的投票进行干扰，比 A4 纸边缘还要显著，进而影响结果。

2. Hough 变换。

对于一个点 (x, y) ，过这个点的直线可以表述成：

$$\rho = x \cos \theta + y \sin \theta$$

其中 ρ 是从原点到直线的有向距离， θ 是直线法向量的辐角。

我们可以把坐标系原点置于图像中心，这样 x 的取值范围是 $[-w/2, w/2]$ ， y 的取值范围是 $[-h/2, h/2]$ 。我们很容易得到 ρ 的取值范围是 $[-\sqrt{w^2 + h^2}/2, \sqrt{w^2 + h^2}/2]$ ， θ 的取值范围是 $[0, 2\pi]$ 。

容易发现 (ρ, θ) 所表示的直线与 $(-\rho, \theta + \pi)$ 所表示的直线是同一直线。所以我们可以将 ρ 的范围减为 $[0, \sqrt{w^2 + h^2}/2]$ 。接下来我们对参数空间进行离散化从而构造我们的投票矩阵。对于每一个边缘点 (x, y) ，枚举所有的 θ ，从而算出对应的 ρ 。如果 $\rho < 0$ ，那么将 (ρ, θ) 变换到 $(-\rho, \theta + \pi)$ 并贡献投票即可。

3. 局部极大值的寻找。

对于上一步的结果，我们需要对结果进行筛选，用其局部极大值来表示我们的结果。首先我们如果用一个阈值来对上一步结果来二值化的话，我们便可以只关注高票的那些参数点了。由于本问题的特殊性，对于一条直线，其对应的高票参数点相对来说隔得非常近；对于不同直线，其对应的高票参数点相对来说隔得较远。所以我们期望通过聚类分析来找到表示一条线的几组参数，并用这些参数中票数最高者作为其局部极大值。

说到聚类分析，我们自然想到的是 k -均值算法，并由于本问题的特殊性，我们可以令 k 为4。但是实际效果是极差的，这是因为 k -均值算法需要照顾到所有的点，所以所聚出来的类会因为噪声的干扰而变得非常难看。在此，我们考虑到问题的特殊性，使用类似 Kruskal 算法的聚类算法。

在图论中，Kruskal 算法通常被用来找最小生成树。我们在这里认为两个点之间的距离小于某一阈值即是同类的点，否则是不同类的点。因此，利用 Kruskal 算法中所用到的并查集数据结构，我们能够在 $O(p^2)$ 的时间复杂度中完成聚类，其中 p 是我们对投票矩阵二值化之后的点数，可以被人为控低。聚类后选出前4大类即可。

4. 角点的求法。

利用 Cramer 法则即可。

对于两条线 (ρ_1, θ_1) 和 (ρ_2, θ_2) ，联立方程组

$$\begin{cases} x \cos \theta_1 + y \sin \theta_1 = \rho_1 \\ x \cos \theta_2 + y \sin \theta_2 = \rho_2 \end{cases}$$

$$\text{记 } A = \begin{vmatrix} \rho_1 & \sin \theta_1 \\ \rho_2 & \sin \theta_2 \end{vmatrix}, B = \begin{vmatrix} \cos \theta_1 & \rho_1 \\ \cos \theta_2 & \rho_2 \end{vmatrix}, C = \begin{vmatrix} \cos \theta_1 & \sin \theta_1 \\ \cos \theta_2 & \sin \theta_2 \end{vmatrix}.$$

$$\text{则 } x = \frac{A}{C}, y = \frac{B}{C}.$$

经观察，只用保留结果仍在图内的 (x, y) 系数即可。

5. 圆的 Hough 变换。

对于一个点 (x, y) ，过这个点的圆可以表述成：

$$(x - x_0)^2 + (y - y_0)^2 = r^2$$

其中 (x_0, y_0) 是圆的圆心， $r > 0$ 是圆的半径。所以参数空间是一个三维的空间。同样对于所有边缘点进行投票枚举，枚举 x_0 和 y_0 ，再通过圆的方程计算出半径 r 的取值。但是如果直接对边缘结果进行 Hough 变换的话结果会受噪声干扰较大。这个时候我们需要对边缘先进行形态学扩张，再进行 Hough 变换。

而且投票的时候要特别注意投票值的选取。对于一组系数 (x_0, y_0, r) ，其贡献应该为 $1/r$ ，对应的意义为这个点对圆所贡献的角度。在二值化筛选的时候就可以根据投票结果（圆完成的程度）来作为阈值（如投票结果 $2\pi\lambda$ 即表示圆完成了 λ ）。

另外，由于我们的参数有三维 (x_0, y_0, r) ，所以我们的参数空间不能被离散化得过细。

6. 圆的局部极大值寻找

和直线局部极大值寻找类似。这里我们再次看出类 Kruskal 算法较 k -均值算法的优越性，即不用监督选类个数。

另外，聚类完毕后，我们要排除掉某些“独轮车”，即过于小的类。

四. 算法代码 (CImg) 实现

见压缩包下代码。通过在解决方案的启动项目中切换两个子任务。

五. 实验结果

见工程文件下的 Dataset1/myresult 和 Dataset2/myresult 中的结果。

其中 Dataset1 的结果均为正确，Dataset2 的结果除了最后一个，其他均为正确。

六. 分析与评价

其实 Hough 变换是一个非常易于理解的数字图像处理过程，代码也不难写出，只是因为整个任务的特殊性以及流程构造的灵活性让正确结果能出来比较难。参数至少有 Canny

算子的边缘阈值、Hough 参数空间离散化的程度、投票矩阵的二值化阈值、同类点对的最大距离等, 综合影响对象检测的效果, 所以基本上大部分的时间都浪费在调参上了。人生苦短, 退一步海阔天空。