机器视觉第五次作业

1、编写霍夫变换实现图像中直线检测的伪码。重要步骤需要加注。

霍夫变换是一种通过将图像空间中的点映射到参数空间（通常使用极坐标表示）来检测图像中的直线或其他形状的算法，其核心思想是利用图像中的边缘信息，通过参数空间的投票机制来发现可能的直线。每个图像中的边缘点都会在霍夫空间中对应一条线，这些线的交点表示原笛卡尔坐标系的图像中可能存在的直线。当在参数空间中多条线投票给同一个参数点时，它们的交点就代表了一个潜在的直线。算法的伪代码如下所示：

输入：图像 Image（经过边缘检测的二值化图像，如 Canny 边缘检测后的图）

输出：检测到的直线集合 Line

1. 定义霍夫空间的参数：

ρ_max = sqrt(width^2 + height^2) // 最大的ρ值等于图像最大长度（对角线长度）

θ_max = 180° // 角度范围从 0°到 180°（斜率为正则θ在 90~180°负则在 0~90°）

// θ为离散角度范围，分成一定的间隔

θ_resolution = 1° // θ取 1° 步长

ρ_resolution = 1 // ρ的步长可以设置为 1 像素

2. 创建累加器（accumulator）数组：

// 累加器是一个二维数组，大小为 [ρ_max / ρ_resolution, θ_max / θ_resolution]，用来存储每个(ρ, θ)组合的投票数，

accumulator = array[ρ_max / ρ_resolution, θ_max / θ_resolution]

3. 对图像 image 中的每个边缘点 (x, y) 执行以下操作：

// 图像中的每个边缘点都会在霍夫空间中投票

for each pixel (x, y) in image:

if image(x, y) == 1: // 判断该点是否是边缘点

for θ_index = 0 to θ_max - 1: //遍历所有角度值

θ = θ_index * θ_resolution // 根据自定义的步长对每一次遍历的角度进行微调

// 通过公式计算该点在霍夫空间中对应的直线在当前角度θ下的ρ值

```
            ρ = x * cos(θ) + y * sin(θ)
            // 将(ρ, θ)转换为（映射到）累加器中对应的索引
            ρ_index = (ρ + ρ_max) / ρ_resolution
            θ_index = θ_index / θ_resolution
            // 累加器增加投票
            accumulator[ρ_index][θ_index] += 1
            // 每个边缘点会在霍夫空间中为每个可能的直线 (ρ, θ) 投一票
```

4. 查找累加器中的局部最大值：

```
   // 查找投票数最多的(ρ, θ)对应的直线参数
   lines = []
   for each ρ_index, θ_index in accumulator:
      if accumulator[ρ_index][θ_index] > threshold:  // 根据设定的阈值筛选直线
         ρ = ρ_index * ρ_resolution - ρ_max
         θ = θ_index * θ_resolution
         // 记录检测到的直线
         lines.append((ρ, θ))
```

5. 输出检测到的直线集合 line：

```
   //对于每个从 4 步骤中找到的直线(ρ, θ)，恢复直线的方程并绘制到原图像上
   for each (ρ, θ) in lines:
      // 恢复直线方程并绘制直线
      a = cos(θ)//直线的方向余弦
      b = sin(θ)//直线的方向正弦
      x0 = a * ρ//直线在 x 轴上的交点
      y0 = b * ρ//直线在 y 轴上的交点
      // 两点确定一条直线
      x1 = x0 + 1000 * (-b)
      y1 = y0 + 1000 * (a)
      x2 = x0 - 1000 * (-b)
      y2 = y0 - 1000 * (a)
```

```
        // 在原图像上绘制直线
        draw_line(x1, y1, x2, y2)
    // 后处理步骤，可以用于优化结果，例如合并重复的直线或精细化调整检测到的直线
        optimized_lines = post_process_lines(lines)
    返回 optimized_lines
```

2、简述霍夫变换实现图像中圆检测的原理

在笛卡尔坐标系中，圆的方程为：$(x-a)^2+(y-b)^2=r^2$，所以将其映射到霍夫坐标系中，需要使用三个参数 a，b，r 来确定一个圆，因此这个对应的霍夫空间为一个三维坐标系，三位霍夫坐标系中的每一个点都能够唯一表示一个二位笛卡尔坐标系中的圆。而对应的笛卡尔坐标系中的圆上的点(x,y)投影到霍夫空间中代表一个圆心，随后与检测直线类似，不断改变半径并进行投票，从而在霍夫空间中能够选择票数最多的点作为最终检测的圆。