山东大学 计算机科学与技术 学院

计算机视觉 课程实验报告

学号：201900130151	姓名： 莫甫龙	
实验题目： 图像的结构 2		

实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

该实验要实现的是基于霍夫变换的图像圆检测，因为可以使用 canny 函数来进行边缘检测，所以对图像进行 canny 边缘检测，得到边缘图(二值图)，然后对得到的二值图上的非 0 点进行遍历， 将其转换为霍夫空间中的该点经过的所有圆。

$$x\_r=x-r\cos\theta \quad y\_r=y-r\sin\theta$$

对此，我们可以用一个三维数组来将圆心的坐标和半径存下来，并记录这样的值有多少个，对于每一个(x_r, y_r, r)，如果被多次访问，那么就代表这个点可以构建出一个以(x_r, y_r)为圆心，以 r 为半径的圆，那么就可以设置一个阈值，只要访问次数大于这个阈值的点，那么就可以以此来画圆。

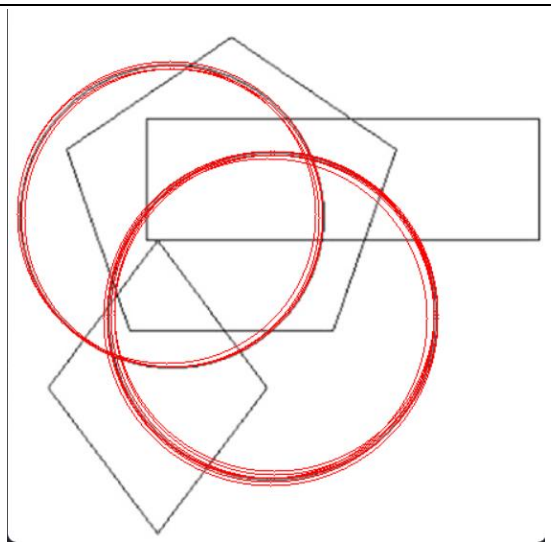
```
cvtColor( src: imge, dst: input, code: COLOR_BGR2GRAY);
GaussianBlur( src: input, dst: input, ksize: Size( _width: 3, _height: 3), sigmaX: 3, sigmaY: 3);
Canny( image: input, edges: edge, threshold1: 60, threshold2: 160);
```

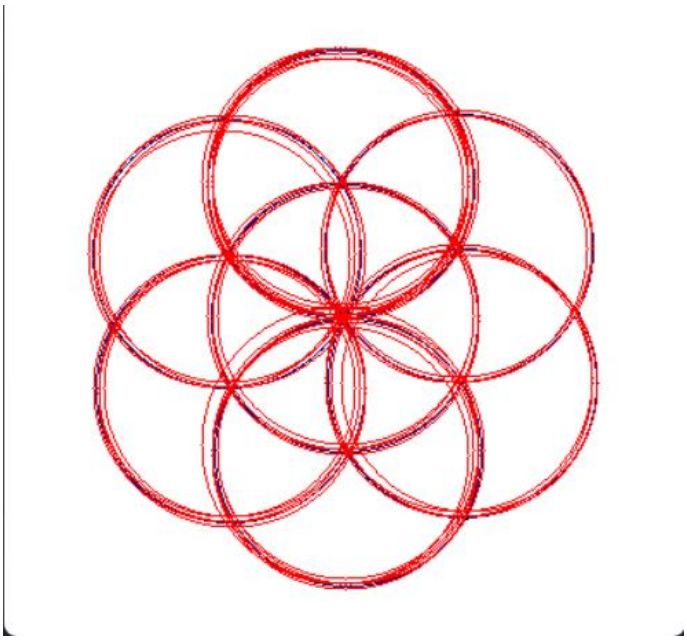
首先将图像转化为灰度图，因为考虑到会有噪声的影响，所以进行一次高斯滤波，然后再进行边缘检测得到边缘图。

```
int theta, y0, x0, y_r, x_r;
for (int x = 0; x < input.rows; x++) {
    for (int y = 0; y < input.cols; y++) {
        if (input.at<uchar>(x, y) != 0)
        for (int r = 95; r < 200; r += 3) {
            y0 = x0 = -1;
            for (theta = 0; theta < 360; theta += 3) {
                x_r = (int) x - r * cos( x: theta * CV_PI / 180);
                y_r = (int) y - r * sin( x: theta * CV_PI / 180);
                if (y_r > 0 && y_r < output.cols && x_r > 0 && x_r < output.rows && y0 != y_r && x0 != x_r) {
                    sum_r[y_r][x_r][r]++;
                    y0 = y_r;
                    x0 = x_r;
                }
            }
        }
    }
}
for (int i = 0; i < output.rows; i+=3) {
    for (int j = 0; j < output.cols; j+=3) {
        for (int r = 95; r < 200; r += 3) {
            if (sum_r[i][j][r] > 30)
                circle( img: output, center: Point(i, j), r, color: Scalar( v0: 0, v1: 0, v2: 255), thickness: 1);
        }
    }
```

接下来就是遍历边缘图中每一个非 0 的点，对于不同的 r 和 theta 来确定不同的圆心，然后将数据保存下来，并且统计次数，最后再进行一次循环来将大于阈值的数据，将它对应的圆画出来，那么这些圆就是图中所含有的圆。

效果图如下：





结果分析与体会：

因为图中的点比较多，所以对于 r 和 theta 的遍历都设置了步长，以此来缩短运行时间。并且从效果图来看，发现会出现重复标记的情况，这个在多次尝试下发现可以通过调阈值来解决，就是十分繁琐。