

计算机视觉 课程实验报告

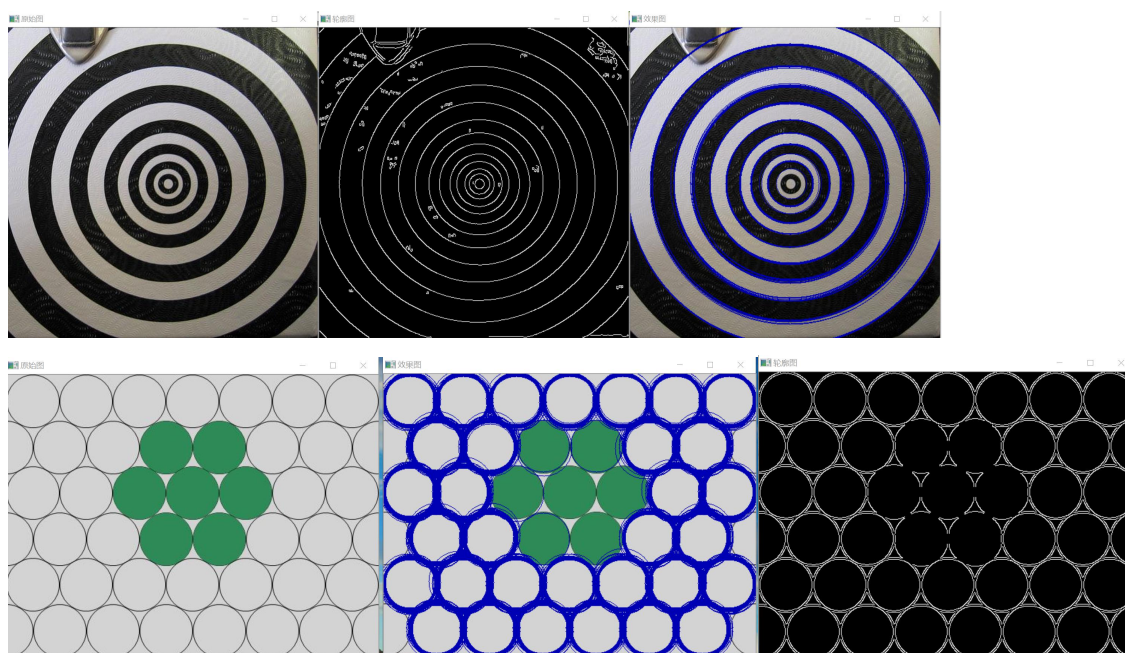
学号：201800130086	姓名：徐鹏博	
实验题目：图像结构 2		
<p>实验过程中遇到和解决的问题：</p> <p>实验 8.1 霍夫变换</p> <ul style="list-style-type: none"> 实现基于霍夫变换的图像圆检测（边缘检测可以用 opencv 的 canny 函数），并尝试对其准确率和效率进行优化实现。 <p>原理：</p> <p>对图像进行边缘检测，得到边缘图(二值图)； 然后对得到的二值图上的非 0 点进行遍历， 将其转换为霍夫空间中的该点经过的所有圆；</p> $\begin{aligned} x, y &\rightarrow a, b, r \\ a &= x - r \cos \theta \\ b &= y - r \sin \theta \end{aligned}$ <p>对于每一个(a, b, r)，如果霍夫空间中的该点被多次访问到，说明存在多个点经过该圆，可以设一个阈值，当满足阈值时认为在原图中存在一个以(a, b)为圆心, r 为半径的圆。</p> <pre> void hough(Mat img,int threshold1,int threshold2,int flag,int min_reduis,int max_reduis) { imshow("原始图", img); Mat BW, edges; cvtColor(img, BW, COLOR_BGR2GRAY); GaussianBlur(BW, BW, Size(3, 3), 3, 3); Canny(BW, edges, threshold1, threshold2); imshow("轮廓图", edges); memset(Houg,0,sizeof(Houg)); int deta_r = 3; int theta, a, b, cura, curb; for (int i = 0; i < edges.rows; i++) { for (int j = 0; j < edges.cols; j++) { if (edges.at<uchar>(i, j) == 0) continue; for (int r = min_reduis; r < max_reduis; r += deta_r) { cura = curb = -1; for (theta = 0; theta < 360; theta += 5) { b = (int) i - r * cos(theta * CV_PI / 180); a = (int) j - r * sin(theta * CV_PI / 180); if (a > 0 && a < img.cols && b > 0 && b < img.rows && cura != a && curb != b) } } } } } </pre>		

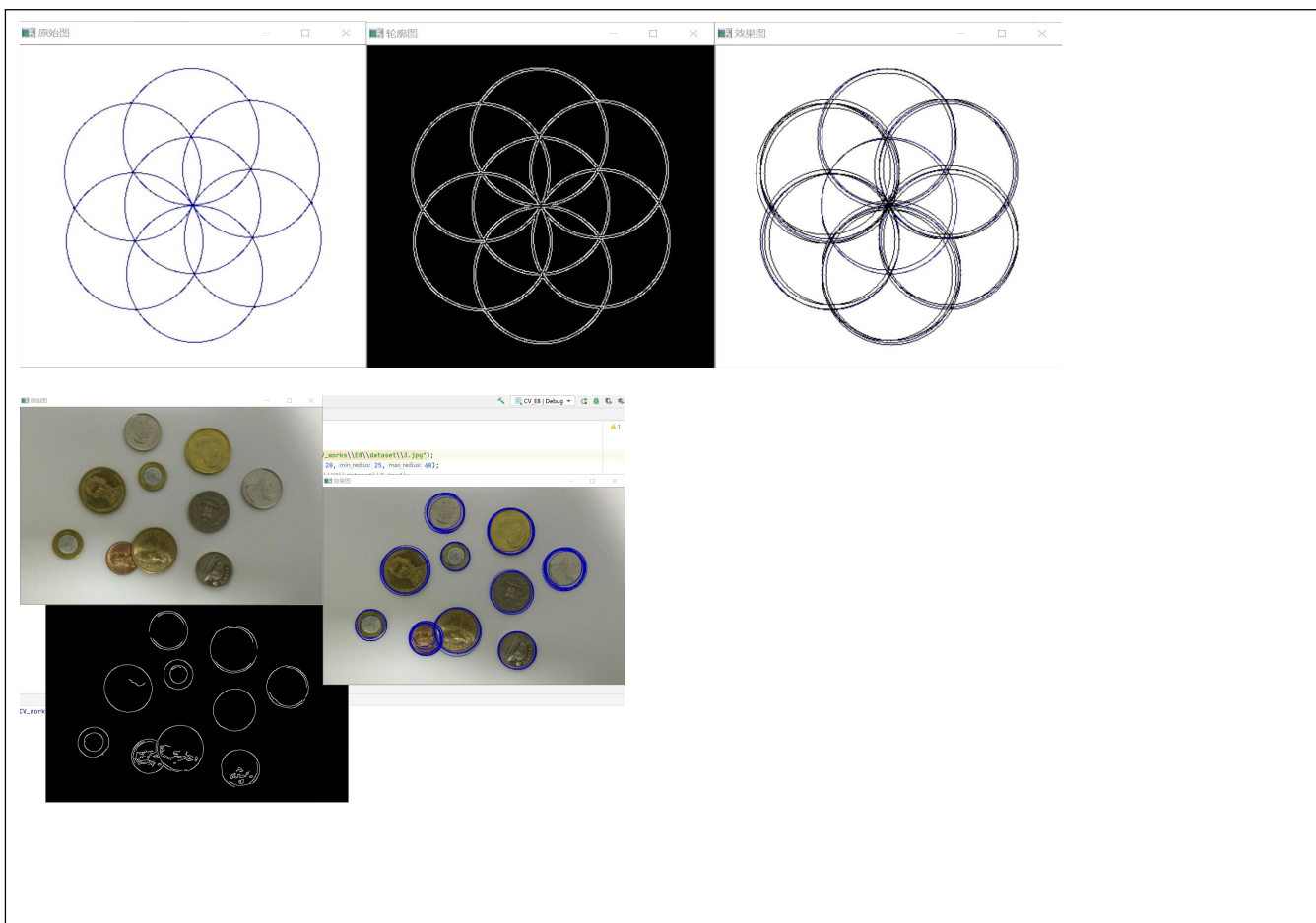
```

    ) {
        Houg[a*maxn*maxn+b*maxn+r]++;
        cura = a;
        curb = b;
    }
}
}
}
for (int i = 0; i < maxn; i+=5) {
    for (int j = 0; j < maxn; j+=5) {
        for (int r = min_reduis; r < max_reduis; r += deta_r) {
            if(Houg[i*maxn*maxn+j*maxn+r]>flag){
                circle(img, Point(i, j), r, Scalar(180, 0, 0), 1);
            }
        }
    }
}
imshow("效果图", img);
}

```

效果图:





结果分析与体会：

由于噪声存在，所以先对得到的边缘图进行了一次高斯滤波处理，又因为原图中的圆的边可能本身具有一定宽度，所以在转换过程中，遍历预设的半径范围时，设置了步长进行控制，避免出现两个几乎重合的圆被认定为不同圆，另外对 **theta** 也设定步长，减少循环量。