

# 浙江大学

## 计算机视觉作业报告

(此模板仅适用于前三个简单的编程作业)

|       |                          |
|-------|--------------------------|
| 作业名称: | Harris Corner Dectection |
| 姓 名:  | 汪鑫                       |
| 学 号:  | 21921164                 |
| 电子邮箱: | wangxin96_2015@163.com   |
| 联系电话: | 18757584990              |
| 导 师:  | 孙建伶                      |

2019 年 12 月 12 日

# 作业名称

(撰写上简明扼要、开门见山，无需废话，文字不在于多)

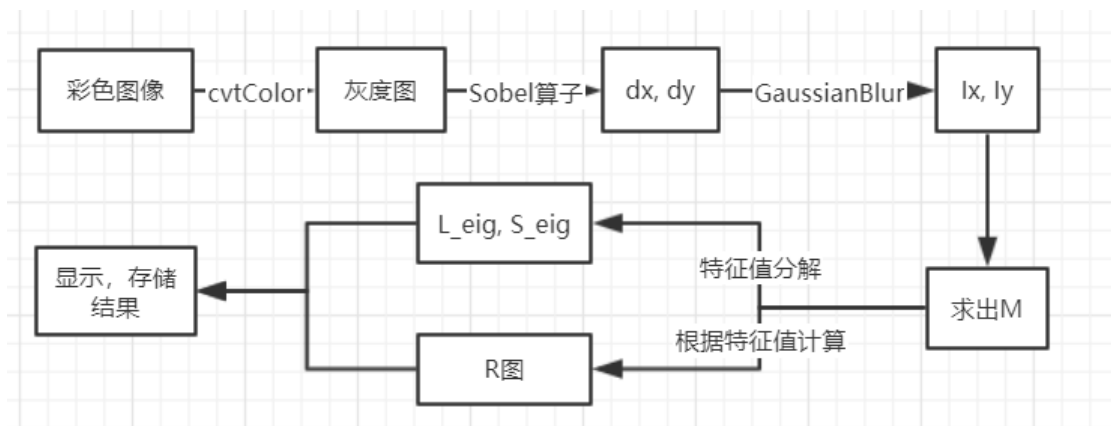
## 一、 作业已实现的功能简述及运行简要说明

1. 将彩色图转换为灰度图；
2. 计算出图像的梯度；
3. 计算出图像的最大特征图/最小特征图/R 图；
4. 在原图上叠加检测结果；

## 二、 作业的开发与运行环境

1. Codeblocks17.12, OpenCv3.0

## 三、 系统或算法的基本思路、原理、流程或步骤等



## 四、 具体如何实现，例如关键（伪）代码、主要用到函数与算法等

1. 计算图像的梯度，直接使用了 sobel 算子，计算更加方便  

```
Mat diff(Mat &src, int para){  
    Mat res;  
    if(para == 0){  
        Sobel(src, res, CV_32FC1, 1, 0, 3, BORDER_DEFAULT);  
    }else{  
        Sobel(src, res, CV_32FC1, 0, 1, 3, BORDER_DEFAULT);  
    }  
    return res;  
}
```
2. 计算矩阵 M 时，直接将步骤 1 中的梯度矩阵与一个高斯模板做卷积  

```
GaussianBlur(dx2, dx2, Size(3, 3), 2.0);
```

3. 计算由于矩阵 M 为 2 阶方阵，有如下结果：

$$\det M = I_x^2 I_y^2 - (I_x I_y)^2, \text{trace}(M) = I_x^2 + I_y^2$$

R 矩阵的计算代码如下：

```
//compute R
Mat detM, traceM, tmp1, tmp2;
multiply(dx2, dy2, tmp1);
multiply(dxdy, dxdy, tmp2);
detM = tmp1 - tmp2;
traceM = dx2 + dy2;
float k = 0.05;
Mat traceM2;
multiply(traceM, traceM, traceM2);
```

4. 求 M 的特征值直接求解方程：

$$(\lambda - I_x^2)(\lambda - I_y^2) - (I_x I_y)^2 = 0$$

结果为：

$$\lambda_{1,2} = 0.5 * \left( (I_x^2 + I_y^2) \pm \sqrt{(I_x^2 + I_y^2)^2 - 4 * I_x^2 I_y^2} \right)$$

结合 3 中的结果，计算特征图的代码为：

```
// compute eig
Mat l_eig, s_eig, sqTmp;
Mat tmp = traceM2 - 4 * detM;
sqrt(tmp, sqTmp);
l_eig = 0.5 * (traceM + sqTmp);
s_eig = 0.5 * (traceM - sqTmp);
```

## 五、 实验结果与分析

见文件夹 result 中的结果

## 六、 结论与心得体会

心得体会：通过自己动手实现了 Harris Corner Detection 算法，我掌握了怎么方便快速的计算图像的梯度和 Harris 算法中的 M 矩阵，自己的编程能力和对图像处理算法有了进一步的提高。

## 七、 参考文献