



ZHEJIANG UNIVERSITY

计算机视觉（本科）实验报告

作业名称 Harris Corner Detection

姓 名 杨逍宇

学 号 3220105453

电子邮箱 1877114891@qq.com

联系电话 13518290755

导 师 潘纲



2024 年 11 月 17 日

1 已实现的功能简述及运行简要说明

1.1 已实现的功能简述:

- (1). 程序运行之后, 会将当前计算机摄像头内容实时显示, 并对当前帧图像做一次 Harris Corner 检测, 并将结果叠加在原来的图像上。
- (2). 还会另开一个回放视频窗口, 播放 R 图。

1.2 运行简要说明:

- (1). 其中可执行文件 `hw_2` 在目录 `build` 下, 在 `ubuntu` 环境下可直接运行。
- (2). 运行程序后, 上述两张图像会显示, 按下 'q' 键可以退出程序。

2 开发与运行环境

本实验使用的软件和工具如下:

- 开发环境: Visual Studio Code on Ubuntu22.04
- 编程语言: C++
- 库: OpenCV 4.7.0
- 构建工具: CMake

3 算法基本思路

Harris 角点检测算法用于识别图像中在所有方向上强度变化显著的点。这些点被称为角点, 在各种计算机视觉任务中非常有用。本作业中涉及以下关键步骤:

- (1). 初始化视频捕获对象。
- (2). 使用 `ksize`, `k`, 和 `threshold_offset` 初始化 `HarrisCornerDetector` 对象。
- (3). 进入视频录制循环:
 - (a) 将输入图像 I 转换为灰度图
 - (b) 使用 Sobel 算子计算梯度 I_x 和 I_y
 - (c) 计算 I_x^2 , I_y^2 , 和 I_{xy}

- (d) 对 I_x^2 , I_y^2 , 和 I_{xy} 应用高斯模糊, 得到 S_x^2 , S_y^2 , 和 S_{xy}
 - (e) 计算结构张量的行列式 $\det M$ 和迹 $\text{trace} M$
 - (f) 使用 $R = \det M - k \cdot (\text{trace} M)^2$ 计算 Harris 响应 R
 - (g) 将 R 归一化到 $[0, 255]$ 范围
 - (h) 显示 R 图
 - (i) 计算自适应阈值 threshold
 - (j) 对 R 应用非极大值抑制以检测角点
 - (k) 返回角点 corners
 - (l) 将处理后的角点叠加在视频上。
 - (m) 显示处理后的帧。
 - (n) 检查用户是否按下 'q' 键, 若按下则退出循环
- (4). 释放摄像头资源。

4 算法实现要点

- **计算梯度:** 使用 Sobel 算子计算图像的 x 和 y 方向梯度:

```
1 Sobel(gray, Ix, CV_64F, 1, 0, ksize_);  
2 Sobel(gray, Iy, CV_64F, 0, 1, ksize_);
```

- **计算结构张量:** 计算结构张量的三个分量:

```
1 Mat Ix2 = Ix.mul(Ix);  
2 Mat Iy2 = Iy.mul(Iy);  
3 Mat Ixy = Ix.mul(Iy);
```

- **高斯模糊:** 对结构张量的三个分量应用高斯模糊:

```
1 GaussianBlur(Ix2, Sx2, Size(ksize_, ksize_), 0);  
2 GaussianBlur(Iy2, Sy2, Size(ksize_, ksize_), 0);  
3 GaussianBlur(Ixy, Sxy, Size(ksize_, ksize_), 0);
```

- **计算 Harris 响应:** 计算结构张量的行列式和迹, 然后计算 Harris 响应:

```
1 Mat detM = Sx2.mul(Sy2) - Sxy.mul(Sxy);  
2 Mat traceM = Sx2 + Sy2;  
3 Mat R = detM - k_ * traceM.mul(traceM);
```

- **计算自适应角点阈值**: 代码使用 `meanStdDev` 函数计算矩阵 `R` 的均值和标准差, 将均值和标准差相加, 再加上一个预定义的偏移量作为角点检测阈值:

```
1 Scalar mean, stddev;
2 meanStdDev(R, mean, stddev);
3 threshold_ = static_cast<int>(mean[0] + stddev[0]) +
    threshold_offset_;
```

- **非极大值抑制**: 对 `R` 进行 NMS 然后根据阈值提取角点:

```
1 void nonMaxSuppression(const Mat& R, vector<Point>& corners)
2 {
3     for (int i = 2; i < R.rows - 2; i++) {
4         for (int j = 2; j < R.cols - 2; j++) {
5             uint8_t value = R.at<uint8_t>(i, j);
6             bool isLocalMax = true;
7             // 5*5 window
8             for (int k = -2; k <= 2; k++) {
9                 for (int l = -2; l <= 2; l++) {
10                     if (k == 0 && l == 0)
11                         continue;
12                     if (R.at<uint8_t>(i + k, j + l) >= value) {
13                         isLocalMax = false;
14                         break;
15                     }
16                 }
17                 if (!isLocalMax)
18                     break;
19             }
20             if (isLocalMax && value > threshold_) {
21                 // cout << "corner value = " << static_cast<int>(
22                     value) << endl;
23                 corners.push_back(Point(j, i));
24             }
25         }
26     }
```

5 实验结果及分析

当设置阈值偏置为零时实验结果如下图所示：

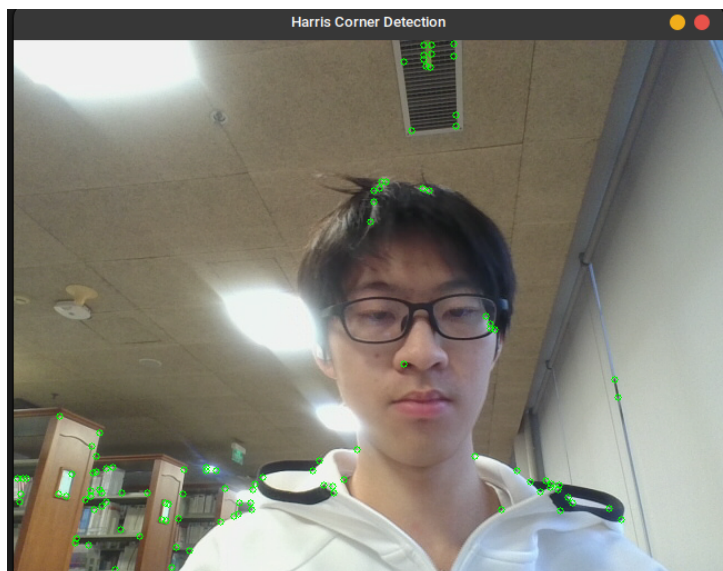


Figure 1: offset=0

从图中可以看到，算法可以较好地检测出角点，并且显示在图像上。相应的 R 图显示如下：



Figure 2: offset=0

可以看到边缘的 R 值表现比平滑区域明显很多。但是在作业开始时我才用了直接设定阈值的方法进行检测，发现角点跳动明显，通过观察 R 图发现 R 图明暗闪烁，推测可能时因为电脑摄像头会进行自适应曝光，于是为了

适应不同曝光情况下的 R 图，我选择了设定自适应的检测阈值，采用 R 图的均值和方差进行计算，并且人为设定检测阈值的偏差值来达到我们想要的效果。结果发现不同的阈值的设定对角点的影响也是较大的：



Figure 3: offset=-10



Figure 4: offset=10

从图像中我们可以看出角点的数量和分布有明显的差别。

6 结论与心得体会

结论

通过本次实验，我们成功实现了 Harris 角点检测算法，并在实时视频流中验证了其有效性。实验结果表明：

- Harris 角点检测算法能够有效地检测出图像中的角点，并且在不同的光照条件下表现出较好的鲁棒性。
- 自适应阈值的使用使得算法能够动态调整检测阈值，从而提高了角点检测的准确性和适应性。
- 算法在普通计算机上能够实现实时处理，具有较高的计算效率。

心得体会

在本次实验中，我学到了以下几点：

- **算法实现：**通过实现 Harris 角点检测算法，我深入理解了图像梯度计算、结构张量构建以及角点响应函数的计算过程。这些知识对理解其他图像处理和计算机视觉算法也有很大帮助。

- **自适应阈值：**自适应阈值的计算方法使我认识到在图像处理过程中，动态调整参数的重要性。通过计算均值和标准差，可以使算法在不同的场景下保持稳定的性能。
- **实时处理：**在实验中实现了实时视频处理，使我体会到算法优化和高效实现的重要性。实时处理要求算法不仅要准确，还要足够快，以满足实时应用的需求。
- **调试与优化：**在实验过程中，遇到了一些调试和优化的问题。通过解决这些问题，我提高了代码调试和性能优化的能力。

总的来说，本次实验不仅让我掌握了 Harris 角点检测算法的实现方法，还让我对图像处理和计算机视觉有了更深入的理解。这些知识和技能将对我今后的学习和研究工作产生积极的影响。