**OpenCV--图像特征-harris**

**【演练目的】**

1.理解Harris 角点检测的概念

2.学习函数：cv2.cornerHarris()，cv2.cornerSubPix()

**【Harris角点检测原理】**

Chris\_Harris 和Mike\_Stephens 早在1988 年的文章《A Combined Corner and Edge Detector》中就已经提出了焦点检测的方法，被称为Harris 角点检测。他把这个简单的想法转换成了数学形式。将窗口向各个方向移动（u，v）然后计算所有差异的总和。表达式如下：



窗口函数可以是正常的矩形窗口也可以是对每一个像素给予不同权重的高斯窗口。

角点检测中要使E (u,v) 的值最大。这就是说必须使方程右侧的第二项的取值最大。对上面的等式进行泰勒级数展开然后再通过几步数学换算得到下面的等式：



 其中:



这里Ix 和Iy 是图像在x 和y 方向的导数。（可以使用函数cv2.Sobel()计算得到）。根据一个用来判定窗口内是否包含角点的等式进行打分。



其中

•   det (M) = 12

•   trace (M) = 1 + 2

• 1 和2 是矩阵M 的特征值

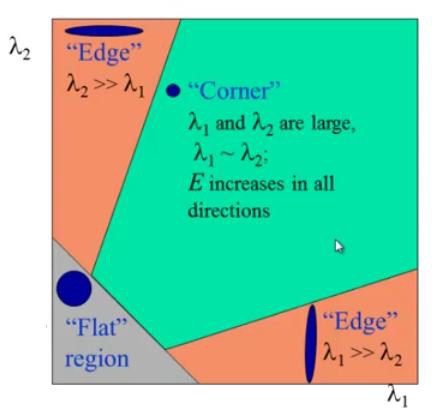
所以根据这些特征中可以判断一个区域是否是角点，边界或者是平面:

•   当1 和2 都小时，|R| 也小，这个区域就是一个平坦区域。

•   当1>>2 或者1<<2，时R 小于0，这个区域是边缘。

•   当1 和2 都很大，并且1～2 中的时，R 也很大，（1 和2 中的最小值都大于阈值）说明这个区域是角点。

可以用下图来表示结论：



所以Harris 角点检测的结果是一个由角点分数构成的灰度图像。选取适当的阈值对结果图像进行二值化就检测到了图像中的角点。

**OpenCV 中的Harris 角点检测**

Open 中的函数cv2.cornerHarris() 可以用来进行角点检测。参数如下：

•   img - 数据类型为float32 的输入图像。

•   blockSize - 角点检测中要考虑的领域大小。

•   ksize - Sobel 求导中使用的窗口大小。

•   k - Harris 角点检测方程中的自由参数，取值参数为[0,04，0.06]。

**亚像素级精确度的角点**

有时需要最大精度的角点检测。OpenCV 提供了函数cv2.cornerSubPix()，它可以提供亚像素级别的角点检测。

**【演练环境】**

Python 3.x

PyCharm

OpenCV-python

**【演练内容】**

理解Harris 角点检测的概念、掌握函数cv2.cornerHarris()，cv2.cornerSubPix()的用法并实现图像的harris角点检测。

**【演练步骤】**

1.打开终端，进入对应文件夹

2.所需数据

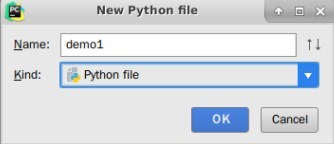
1. opencv16.zip

解压数据

3.打开PyCharm开发工具，新建项目，取名为Harris

**OpenCV 中的Harris 角点检测**

4.在Harris项目下新建python文件，右键-【New】-【Python File】，取名为demo1



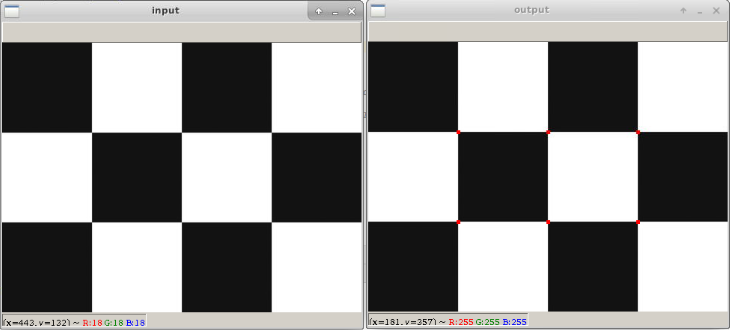
下面开始编写图像角点检测的代码。

5.在demo1中编写代码，首先用opencv读取一张图片，并将读取的图片灰度化，进行角点检测。

demo1.py的完整代码如下：

1. **import** cv2
2. **import** numpy as np
3. filename = 'data/in.jpg'
4. img = cv2.imread(filename)
5. cv2.imshow('input',img)
6. gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)
7. gray = np.float32(gray)
8. # 输入图像必须是float32，最后一个参数在0.04 到0.06 之间
9. dst = cv2.cornerHarris(gray,2,3,0.04)
10. # 返回角点检测结果
11. dst = cv2.dilate(dst,None)
12. # 设置标注角点颜色为红色
13. img[dst>0.01\*dst.max()]=[0,0,255]
14. cv2.imshow('output',img)
15. cv2.waitKey()

右键运行demo1.py，结果如下：

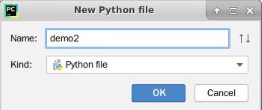


可以看到，输出图像中Harris 角点用红色像素标出。

**亚像素级精确度的角点检测**

OpenCV提供了函数cv2.cornerSubPix()，它可以提供亚像素级别的角点检测。

6.在Harris项目下新建python文件，右键-【New】-【Python File】，取名为demo2



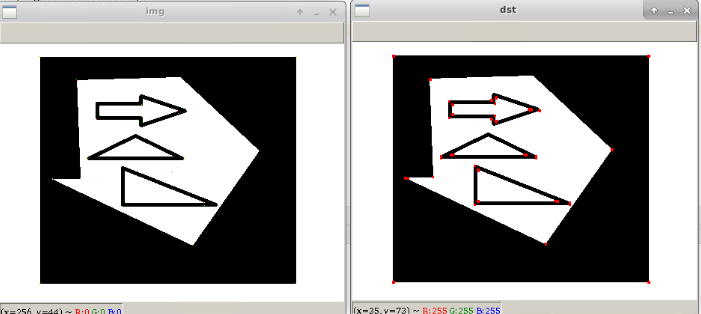
7.首先要找到Harris角点，然后将角点的重心传给这个函数进行修正。

Harris 角点用红色像素标出，绿色像素是修正后的像素。在使用这个函数时，要定义一个迭代停止条件，当迭代次数达到或者精度条件满足后迭代就会停止，同样需要定义进行角点搜索的邻域大小。

demo2.py的完整代码如下：

1. **import** cv2
2. **import** numpy as np
3. filename = 'data/in.png'
4. img = cv2.imread(filename)
5. gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)
6. gray = np.float32(gray)
7. # 输入图像必须是float32，最后一个参数在0.04 到0.06 之间
8. dst = cv2.cornerHarris(gray,2,3,0.06)
9. # 返回角点图片
10. dst = cv2.dilate(dst,None)
11. # 将角点标注为红色
12. img[dst>0.01\*dst.max()]=[0,0,255]
13. cv2.imshow('dst',img)
14. img = cv2.imread(filename)
15. gray = cv2.cvtColor(img,cv2.COLOR\_BGR2GRAY)
16. # find Harris corners
17. gray = np.float32(gray)
18. dst = cv2.cornerHarris(gray,2,3,0.06)
19. dst = cv2.dilate(dst,None)
20. ret, dst = cv2.threshold(dst,0.01\*dst.max(),255,0)
21. dst = np.uint8(dst)
22. ret, labels, stats, centroids = cv2.connectedComponentsWithStats(dst)
23. criteria = (cv2.TERM\_CRITERIA\_EPS + cv2.TERM\_CRITERIA\_MAX\_ITER, 100, 0.001)
24. #Python: cv2.cornerSubPix(image, corners, winSize, zeroZone, criteria)
25. # 返回值由角点坐标组成的一个数组（而非图像）
26. corners = cv2.cornerSubPix(gray,np.float32(centroids),(5,5),(-1,-1),criteria)
27. res = np.hstack((centroids,corners))
28. #np.int0 可以用来省略小数点后面的数字（四舍五入）。
29. res = np.int0(res)
30. img[res[:,1],res[:,0]]=[0,0,255]
31. img[res[:,3],res[:,2]] = [0,255,0]
32. cv2.imshow('img',img)
33. cv2.waitKey()

 右键运行demo2.py，结果如下：



以上！