

## Lab Practice

### 1、 Harris corner detector

*Harris* operator is a simple point feature extraction operator. The matrix  $M$ , inspired by autocorrelation function in signal processing, associated with autocorrelation function which is given. The eigenvalue of the  $M$ -matrix is the first-order curvature of the autocorrelation function. If both curvature values are high, the point is considered to be the eigenpoint. In order to eliminate the influence of noise on corner detection, a Gaussian filter can be used to smooth the image.

### Create a new Harris. Py file

(1) Used to calculate the response function of the corner detector:

```
1  from scipy.ndimage import filters
2  from numpy import *
3  from pylab import *
4  def compute_harris_response(im,sigma=3):
5
6      #计算导数
7      imx=zeros(im.shape)
8      filters.gaussian_filter(im,(sigma,sigma),(0,1),imx)
9      imy=zeros(im.shape)
10     filters.gaussian_filter(im,(sigma,sigma),(1,0),imy)
11     Wxx=filters.gaussian_filter(imx*imx,sigma)
12
13     #计算harris矩阵分量
14     Wxy=filters.gaussian_filter(imx*imy,sigma)
15     Wyy=filters.gaussian_filter(imy*imy,sigma)
16
17     #计算矩阵的特征值和迹
18     Wdet=Wxx*Wyy-Wxy**2
19     Wtr=Wxx+Wyy
20
21     return Wdet/Wtr
22
```

(2) To obtain the descending order of the response values of all candidate pixel points and corner points (**min\_dist=6,threshold=0.1**):

```
def get_harris_points(harrisim,min_dist=6,threshold=0.1):
    """从一幅harris响应图像中返回角点。
    min_dist为分割角点和图像边界的最少像素数目"""
    conner_threshold=harrisim.max()*threshold
    harrisim_t=(harrisim>conner_threshold)*1

    #得到候选点的坐标
    coords=array(harrisim_t.nonzero()).T

    #以及它们的Harris响应值
    candidate_values=[harrisim[c[0],c[1]] for c in coords]

    #对候选点按照Harris响应值进行排序
    index=argsort(candidate_values)

    #将可行点的位置保存到数组中
    allowed_locations=zeros(harrisim.shape)
    allowed_locations[min_dist:-min_dist,min_dist:-min_dist]=1

    #按照min_distance原则，选择最佳harris点
    filtered_coords=[]
    for i in index:
        if allowed_locations[coords[i,0],coords[i,1]]==1:
            filtered_coords.append(coords[i])
            allowed_locations[(coords[i,0]-min_dist):(coords[i,0]+min_dist),
                             (coords[i,1]-min_dist):(coords[i,1]+min_dist)]=0
            #此处保证min_dist*min_dist只有一个harris特征点
    return filtered_coords
```

(3) Display the corner points in the image:

```
53 def plot_harris_points(image,filtered_coords):
54     """绘制图像中检测到的角点"""
55
56     figure()
57     gray()
58     imshow(image)
59     plot([p[1] for p in filtered_coords],[p[0] for p in filtered_coords],'+')
60     axis('off')
61     show()
62
```

**In the same directory as Harris.py, create a new practice.py file**

(4) Call the established function and run the following command:

```
1  from PIL import Image
2  from numpy import *
3  import harris
4  from pylab import *
5  from scipy.ndimage import filters
6
7  im=array(Image.open('a.jpg').convert('L'))
8  harrisim=harris.compute_harris_response(im)
9  filtered_coords=harris.get_harris_points(harrisim)
10 harris.plot_harris_points(im,filtered_coords)
11
```

(5) Examples are shown below:



## 2、 Call Harris corner detector in the PCV module

### (1) Steps to install the PCV module library:

[https://blog.csdn.net/weixin\\_42578378/article/details/88617207](https://blog.csdn.net/weixin_42578378/article/details/88617207)

### (2) Use the Harris corner detector to detect corners

(min\_dist=6,threshold=[0.01,0.5,0.1]):

```
from pylab import *
from PIL import Image
from PCV.localdescriptors import harris

# 读入图像
im = array(Image.open('a.jpg').convert('L'))

# 检测harris角点
harrisim = harris.compute_harris_response(im)

# Harris响应函数
harrisim1 = 255 - harrisim

figure()
gray()

# 画出Harris响应图
subplot(221)
imshow(harrisim1)
print(harrisim1.shape)
axis('off')
axis('equal')

threshold = [0.01, 0.05, 0.1]
for i, thres in enumerate(threshold):
    filtered_coords = harris.get_harris_points(harrisim, 6, thres)
    subplot(2, 2, i + 2)
    imshow(im)
    print(im.shape)
    plot([p[1] for p in filtered_coords], [p[0] for p in filtered_coords], '*')
    axis('off')

show()
```

### (3) Examples are shown below:

