

计算机视觉 实验二 报告

张俊华 16030199025

郁张超 16030140077

一、实验内容

特征检测与匹配的目标是识别一个图像中的关键点与另一个图像中的对应点之间的配对。在此实验中，你将编写代码以检测图像中的特征点（对于平移、旋转和照明具有一定的不变性），并在另一个图像中找到最佳匹配特征。

实验分工：

- 张俊华：TODO 1、3、5、7
- 郁张超：TODO 2、4、6、8

二、实验环境

Python 3.7.1 (v3.7.1:260ec2c36a, Oct 20 2018, 14:05:16) [MSC v.1915 32 bit (Intel)] on win32

PyCharm 2018.2.4 Build #PY-182.4505.26, built on September 19, 2018 Windows 10 10.0

三、实验过程

为图像中每个像素计算 Harris 强度函数与方向

由 Harris 矩阵定义可知，要计算得到每个像素点的 Harris 强度，首先需要计算出该点处的方向导数

根据题目要求，使用 Sobel 算子进行图像滤波，以获得 x 和 y 方向的导数：

```
1 dx = np.zeros(srcImage.shape[:2])
2 scipy.ndimage.sobel(srcImage,axis=0,output=dx,mode='reflect')
3 dy = np.zeros(srcImage.shape[:2])
4 scipy.ndimage.sobel(srcImage,axis=1,output=dy,mode='reflect')
```

之后需要为方向导数的运算结果增加权重，使用标准差为 0.5 的高斯卷积核进行卷积

```
1 gxx = scipy.ndimage.gaussian_filter(dx*dx,0.5)
2 gxy = scipy.ndimage.gaussian_filter(dx*dy,0.5)
3 gyy = scipy.ndimage.gaussian_filter(dy*dy,0.5)
```

最终 Harris 矩阵计算方法如下：

```
1 harrisImage = (gxx * gyy - gxy*gxy - 0.1*(gxx+gyy)*(gxx+gyy))
```

Harris 强度的方向即为 dx、dy 的反正切

```

1 for i in range(height):
2     for j in range(width):
3         orientationImage[i][j] = np.arctan2(dx[i][j], dy[i][j]) * 180 / np.pi

```

计算布尔矩阵，指示每个像素是否是局部最大值

```

1 maxImage = scipy.ndimage.filters.maximum_filter(harrisImage, size=(7,7))
2     height, width = harrisImage.shape[:2]
3     for i in range(height):
4         for j in range(width):
5             if maxImage[i][j] == harrisImage[i][j]:
6                 destImage[i][j] = True
7     return destImage

```

使用 `scipy.ndimage.filters.maximum_filter` 函数对图像进行滤波，滤波后，每个像素的结果即为 7*7 窗口中的灰度最大值，之后，使用滤波结果与原始图像进行比较，若像素值与原始图像对应位置的像素值相同，该像素即为局部最大值

根据像素的Harris强度与是否局部最大值生成特征点集合

```

1 f.size = 10
2 f.pt = (x,y)
3 f.angle = orientationImage[y][x]
4 f.response = harrisImage[y][x]
5 features.append(f)

```

按照要求，填写特征点元素的 `size`、`pt`、`angle`、`response` 字段属性。并将特征点添加到 `features` 集合中

简单描述符 `SimpleFeatureDescriptor` 实现

简单的描述符 `SimpleFeatureDescriptor` (TODO 4)，它是 5×5 邻域中的像素强度值（即灰度值）

```

1         for r in range(5):
2             for l in range(5):
3                 try:
4                     desc[i][r*5+l] = grayImage[y+r-2][x+l-2]
5                 except Exception:
6                     desc[i][r * 5 + l] = 0

```

简单描述符 `SimpleFeatureDescriptor` 是一个 25 维的向量，因为直接存放了邻域的灰度值，因此只需要在 for 循环中进行循环赋值即可。

若特征点在图像边界处，邻域中的像素位于图像外时，会触发数组越界异常，循环中捕捉到此异常后，直接将该点的强度值按照 0 处理

MOPS描述符的简化版本 `MOPSFeaturesDescriptor`

该任务的目的是需要在特征点周围的 40*40 像素区域进行子采样，因为待匹配的特征点可能发生了旋转，故需要将特征方向统一。

变换矩阵是由以下几个矩阵组合得到：

平移 (T1)、旋转 (R)、缩放 (S) 和平移 (T2) 由于 `transformations` 中的函数进行的均为三维变换，得到变换矩阵之后需要对其进行提取，得到二维变换矩阵

```
1 angle = np.deg2rad(f.angle)
2 T1 = transformations.get_trans_mx(np.array([-x, -y, 0]))
3 R = transformations.get_rot_mx(0, 0, -angle)
4 S = transformations.get_scale_mx(.2, .2, 0)
5 T2 = transformations.get_trans_mx(np.array([4, 4, 0]))
6 four_x_four = np.dot(np.dot(np.dot(T2, S), R), T1)
7
8 transMx = four_x_four[0:2, [0, 1, 3]]
```

规范化为零均值和单位方差

直接调用 `np.std`、`np.mean` 函数进行归一化，`np.std` 求得矩阵的均值，`np.mean` 求得矩阵的方差，矩阵减去方差除以均值即得到归一化结果。

需要注意的是，如果方差非常接近零（幅度小于 10^{-5} ），那么应该返回一个全零向量以避免除零错误

```
1 std = np.std(destImage)
2     if std ** 2 < 1e-10:
3         desc[i] = (np.zeros(np.shape(destImage))).flatten()
4     else:
5         desc[i] = ((destImage - np.mean(destImage)) / std).flatten()
```

SSDFeatureMatcher

```
1 dists = spatial.distance.cdist(desc1, desc2)
2 mins = np.argmin(dists, axis=1)
3 for i in range(len(mins)):
4     m = cv2.DMatch(_queryIdx=i, _trainIdx=mins[i],
5     _distance=np.amin(dists[i]))
6     matches.append(m)
```

RatioFeatureMatcher

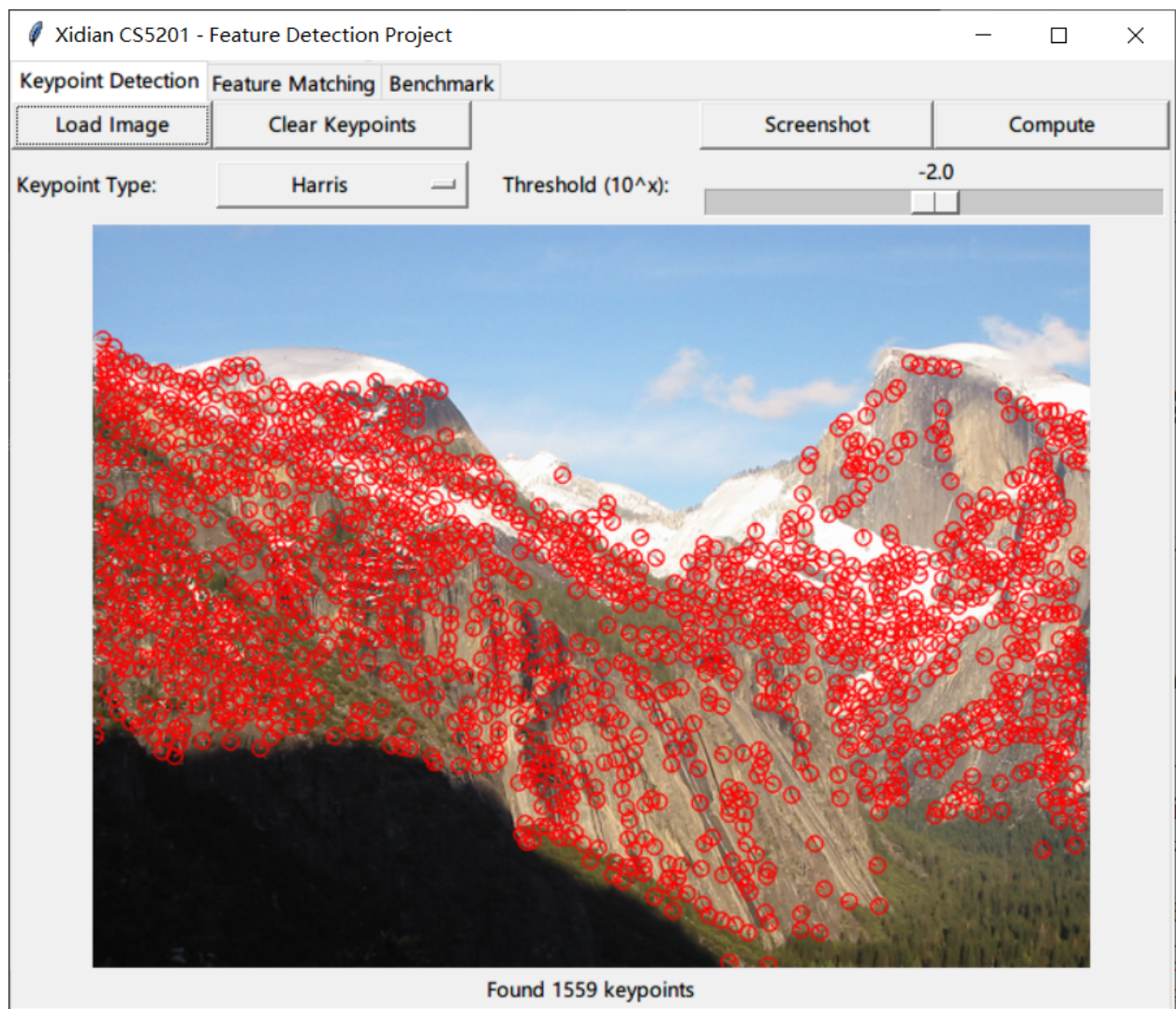
```
1 dists = spatial.distance.cdist(desc1, desc2)
2 mins = np.argmin(dists, axis=1)
3 for i in range(len(mins)):
4     distsi = dists[i]
5     bestMatch = mins[i]
6     bestDist = np.amin(distsi)
7     # alter distsi to change the previous min to a really big number
8     distsi[mins[i]] = float('inf')
9     secondBestMatch = np.argmin(distsi)
```

```
10     secondBestDist = np.amin(dists[i])
11
12     ratioDist = bestDist / float(secondBestDist)
13
14     m = cv2.DMatch(_queryIdx=i, _trainIdx=mins[i], _distance=ratioDist)
15     matches.append(m)
```

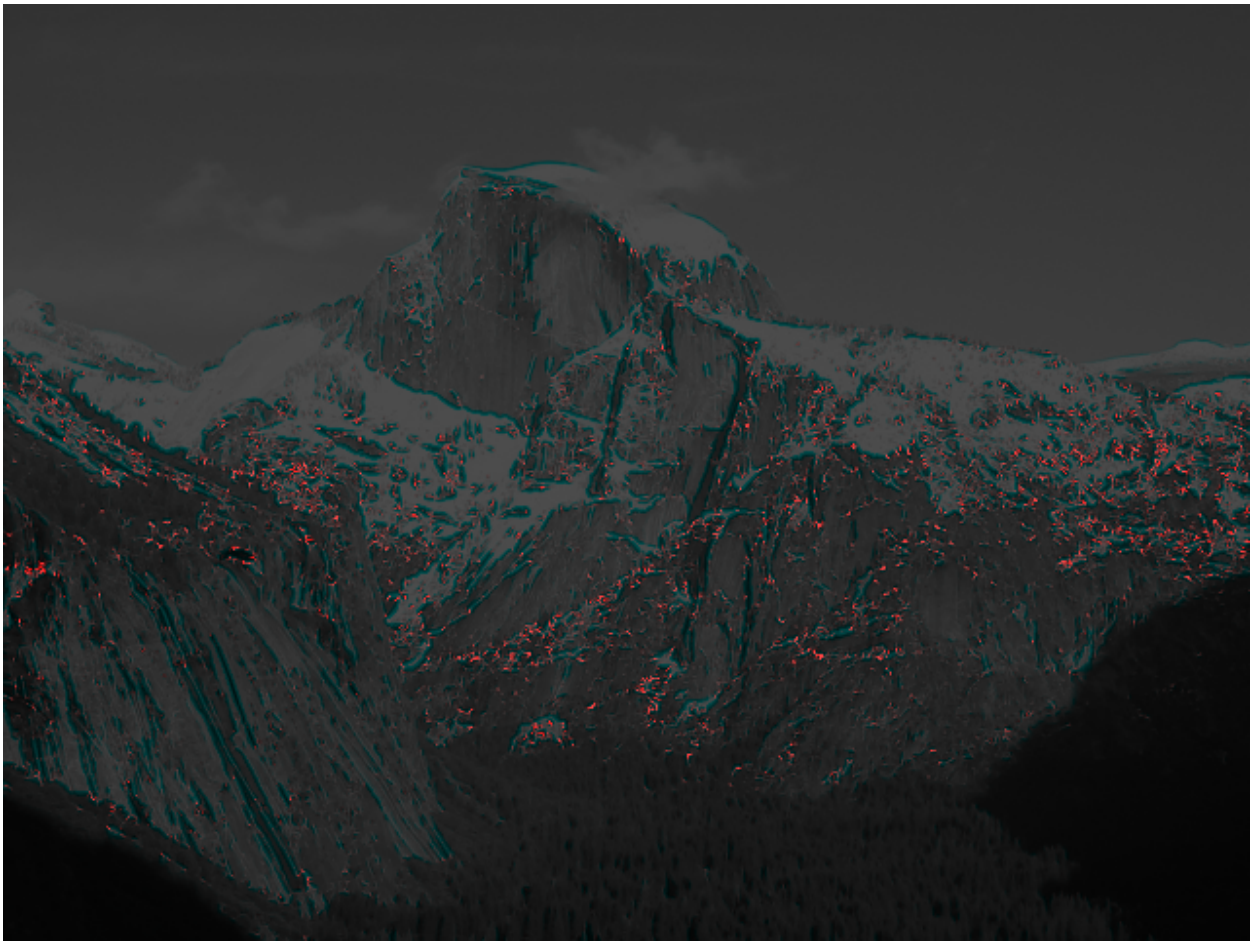
四、实验结果

- 角点检测

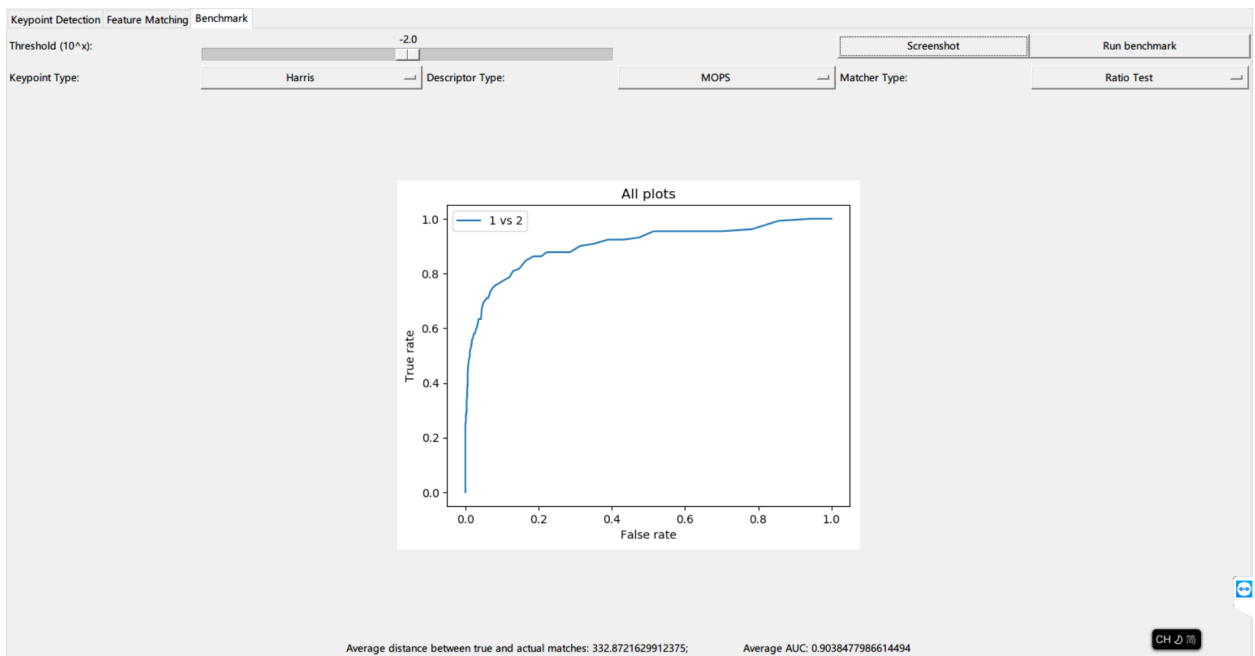
使用测试样例集中的 `yosemite1.jpg` 来进行检测：

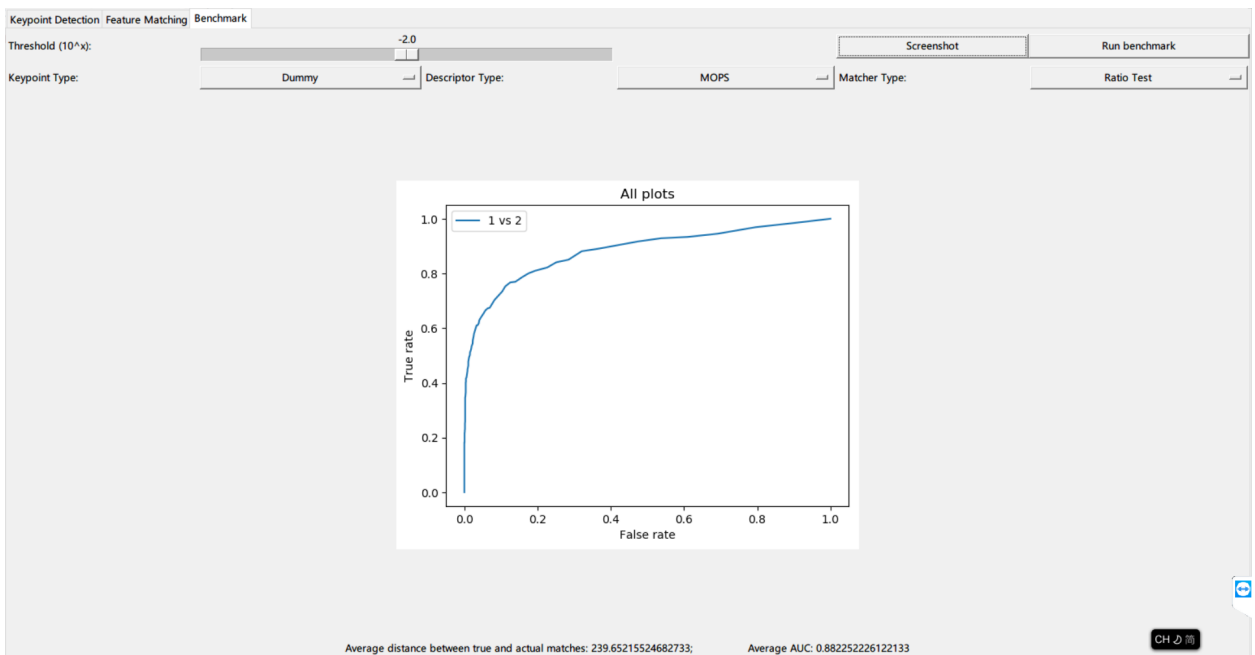
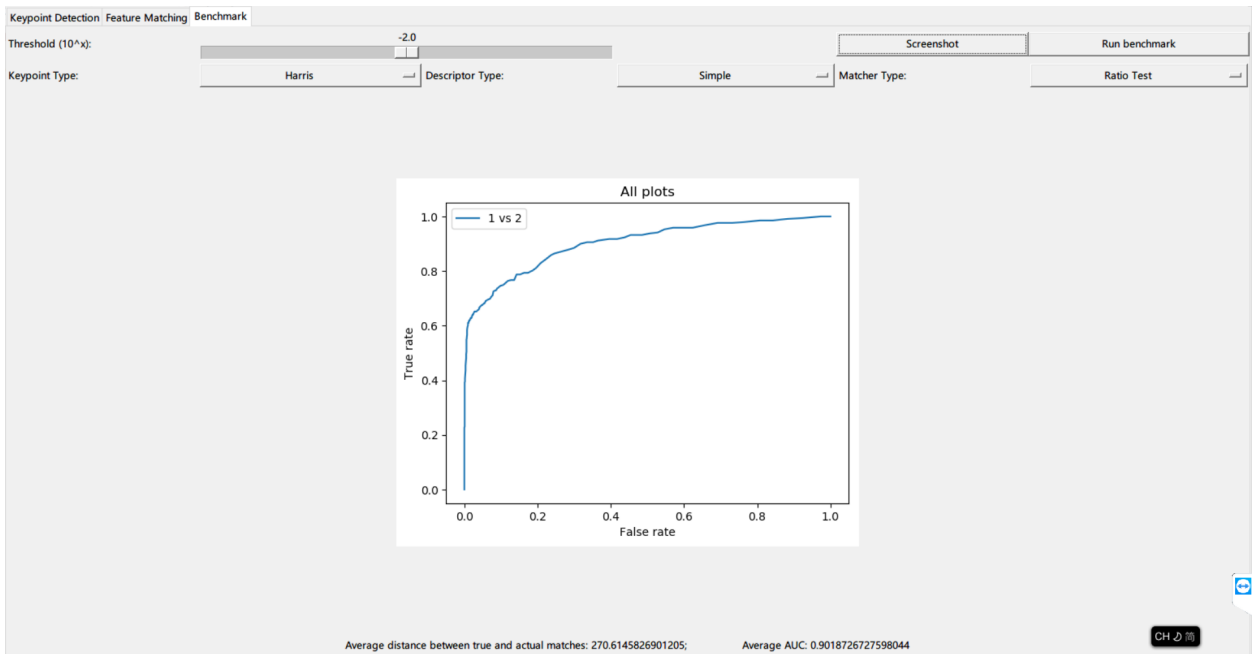


对应的 harris.png 为：



- ROC曲线和AUC







五、实验心得

这次的实验是我们两人（张俊华16030199025）（郁张超 16030140077）共同完成的实验。本次实验中所有的 TODO 任务基本上都是两个人共同完成的，与一个人做实验不同，两个人共同实验，更能发现完全不同的思路。遇到解决不了的问题两个人一起讨论，往往能更快得以解决。

此次实验的目的是完成图像特征检测与匹配，通过计算Harris角点以及实现特征描述符来实现匹配工作。其中，一个好的特征描述符很大程度上影响匹配的效果，从简单特征描述符与MOPS简化版本的实现效果比较便可知这一点。通过动手实践让我弥补了理论课程中知识的漏洞与不足，同时让我们认识到了想要实现一个好的特征匹配需要考虑很多方面，例如特征描述符的尺度不变特性等。

最终的结果使用 GUI 的方式进行呈现，使得我们更加清晰的观察到了不同特征匹配方式的效果，对课本的知识有了更加深刻的印象，使我们受益匪浅。

