实验二 特征检测与匹配

1. 实验目的

实现Harris角点检测，MOPS，Simple特征描述，以及SSD，RatioTest特征匹配。

1. 实验环境及分工
2. 环境：基于python3.7.4
3. 分工：全部独立完成
4. 实验内容

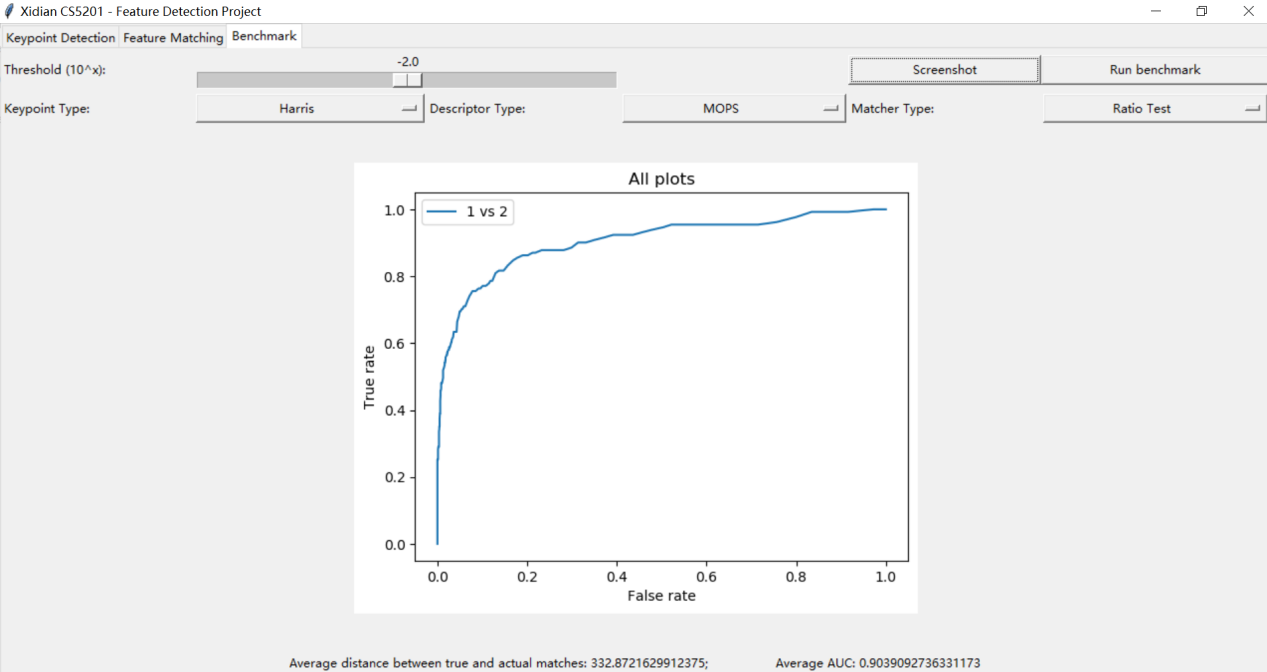
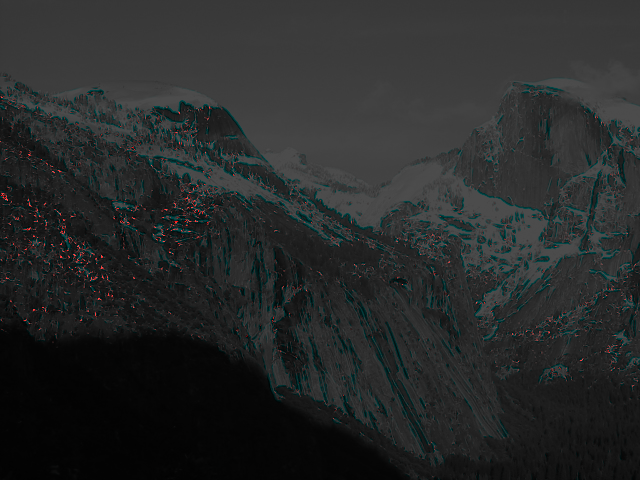
本次实验包括三大部分：特征检测，特征描述，特征匹配

1. 特征检测：本次实验中实现了Harris算子，Harris算子的实现分为 三小部分
   1. 计算每个像素处的角点响应强度大小及方向。首先通过Sobel算子求得x方向和y方向上的导数。根据MOPS论文中所述，为了防止混叠，在较高的高斯金字塔上进行采样，本实验中使用σ=0.5的高斯核处理，求得H矩阵的四个参数，带入公式得到每一点的角点响应强度。方向由该点处的梯度近似给出，注意要转为角度才能通过test.py。
   2. 检测每个点是否是局部最大值，针对TODO2采用了简化处理，只判断每个点是否是7\*7窗口的极大值即可。后续我实现了非极大值抑制，性能提高27%
   3. 将特征点的角点响应强度，方向，坐标等信息放进cv2.KeyPoint对象列表中，返回特征点列表。
2. 特征描述：本次实验中分别实现了SimpleFeatureDescriptor和MOPS特征描述。
   1. SimpleFeatureDescriptor：先给矩阵加一圈2\*2的白边，以防越界，然后取每个点周围5\*5的区域，并以行序为主序构成描述向量。
   2. MOPS实现需要用到cv2的仿射变换函数，仿射变换用到的矩阵由四部分矩阵乘法得到：分别是平移到原点的矩T1 = np.array([[1,0,-x],[0,1,-y],[0,0,1]]) ，旋转到水平的矩阵R = np.array([[math.cos(angle),-math.sin(angle),0],[math.sin(angle),math.cos(angle),0],[0,0,1]])缩放的矩阵S = np.array([[1/5,0,0],[0,1/5,0],[0,0,1]])

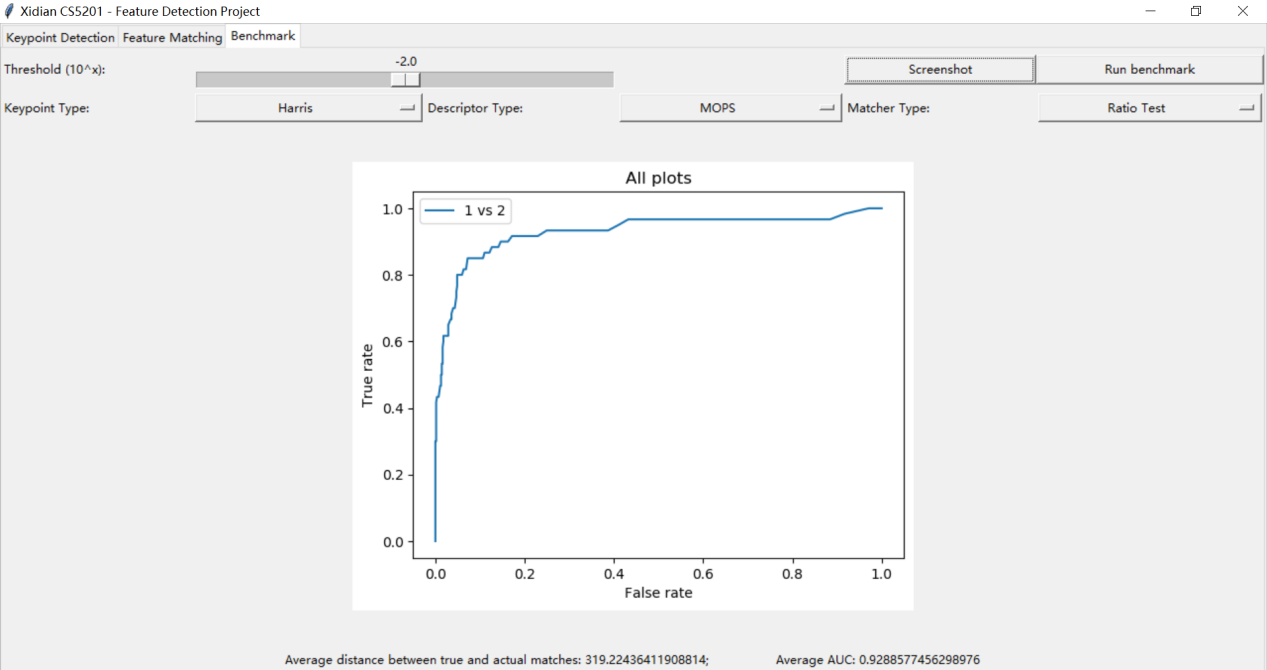
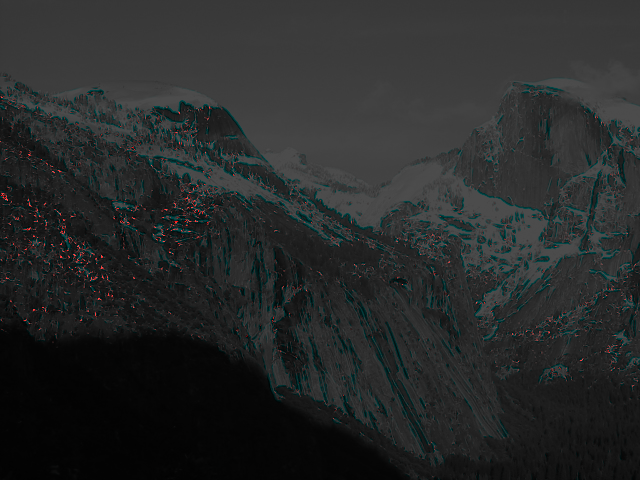
平移到中心的矩阵T2 = np.array([[1,0,4],[0,1,4],[0,0,1]])。这四部分进行矩阵乘法的结果作为仿射变换的矩阵，将变换的结果按行序为主序展成列表，得到特征描述符。

1. 特征匹配：本次实验中实现了两种特征匹配SSD和RatioTest，二者的区别是SSD是求到其他特征点距离的平方和，二RatioTest是选择其中最大的两个距离的平均值。
2. 加分项

实现了自适应非极大值抑制，在选择区域内最大点的时候，从半径等于图像大小开始，枚举半径，将半径最大的1250(论文中是500，实测出来1500左右的AUC比较好)点选出来，作为基础的特征点。通过在yosemite数据集上的测试，性能提升27%。

**未加极大值抑制**

**AUC = 0.9039092736**

**加入自适应非极大值抑制 ANSM**

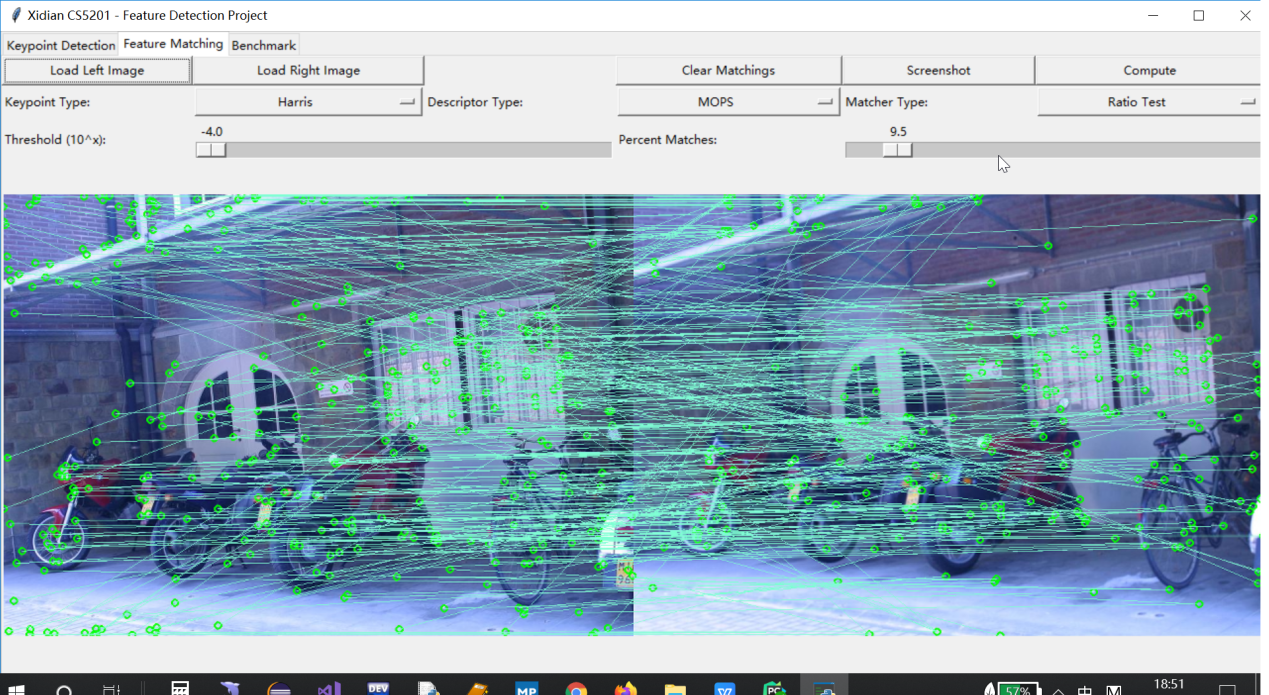
**AUC = 0.928857745**

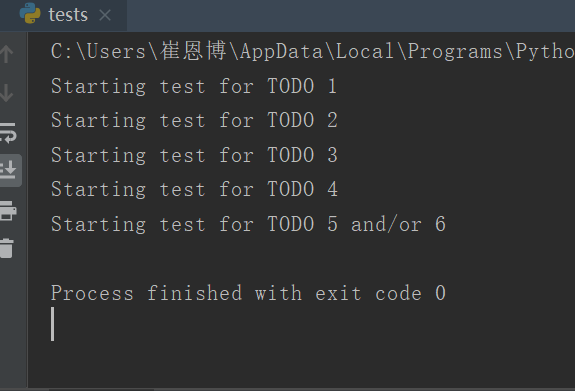
1. **实验结果**
2. **各种搭配的AUC曲线，放在文件夹中。**

**通过对各个中搭配的比较，可知对于特征检测，Harris效果好于Dummy，对于特征描述MOPS好于SIMPLE，对于特征匹配，RatioTest好于SSD。**

1. **对于特征检测：由于Harris选择的是在各个方向上都变化较大的点，并给出了角点响应的强度及方向。**
2. **对于特征描述，简单的特征描述没有将窗口内的像素放在同一个坐标系下进行，并且没有考虑旋转不变性，所以性能不如MOPS**

**(3) RatioTest相比于SSD的改进在于距离的定义，RatioTest中的距离定义为某个点到最佳匹配点的距离/到次佳匹配点的距离，这样不好的匹配会给出较大的响应。之后通过阈值过滤可以留下更多正确的匹配(针对有多个匹配的问题)**

****

**test文件全部通过。**

**17069130005**

**崔恩博**