

SIFT 图象拼接实验报告

姓名：王道烜

学号：2015011006

班级：无 52

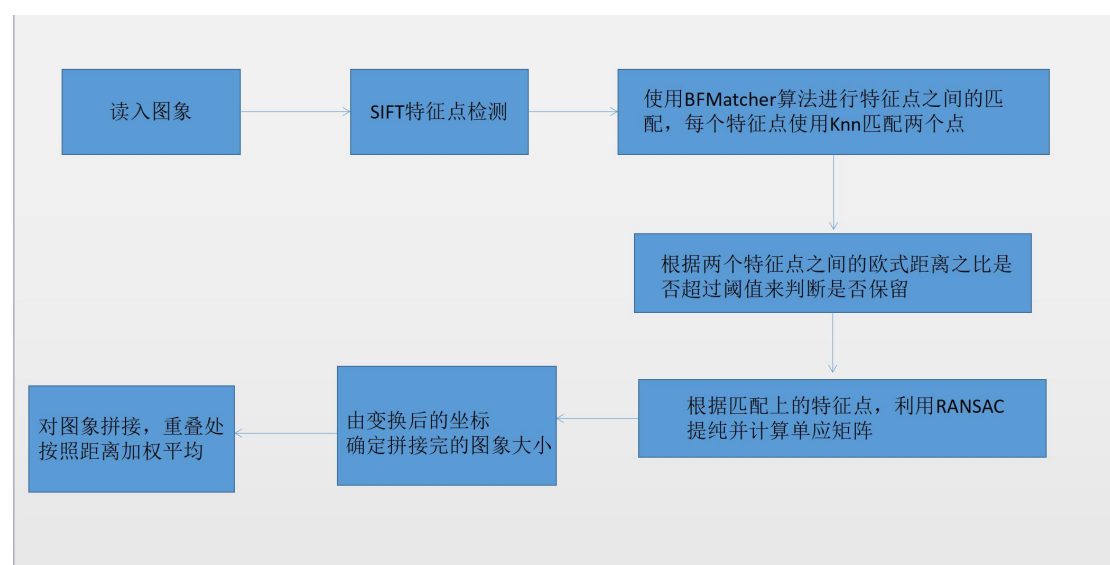
一、实验原理

本次实验主要是利用 SIFT 算法对图象的特征点进行提取，然后先利用 knn 方法对特征点进行预筛选，然后利用 RANSAC 算法进行提纯并求得单应矩阵，最后将单应矩阵作用到图像上，进行变换，然后在将图象拼接起来，最终组成完整图象。

二、实验环境

Ubuntu16.04 + opencv_contrib 3.4.1 + numpy

三、实验流程图



四、实验步骤

(1) 首先利用 opencv 库中的函数对图象进行特征点提取，对实验图像可以得到如下结果：



可以看出对图象的特征提取的非常充分。

(2) 然后使用 `opencv` 中的类 `BFMatcher()` 中的 `Knnmatch()` 方法对两幅图片之间的特征点进行匹配。这里就第一张图中的每一个特征点，在第二张图中寻找两个和其欧式距离最近的两个点，并将匹配的结果保存在列表中。

(3) 对得到的匹配结果进行筛选。这里使用的方法是，测试两个最近的距离之间的比值是否满足一个阈值，如果小者比大者的比值小于某一个阈值的话，就可以认为这一对匹配点是一对比较好的匹配点，保存在 `good` 列表中。这个阈值论文作者选择 **0.8**，本人在本次实验中选择 **0.5** 效果比较好。经过这样的筛选之后，得到的匹配结果如下所示：



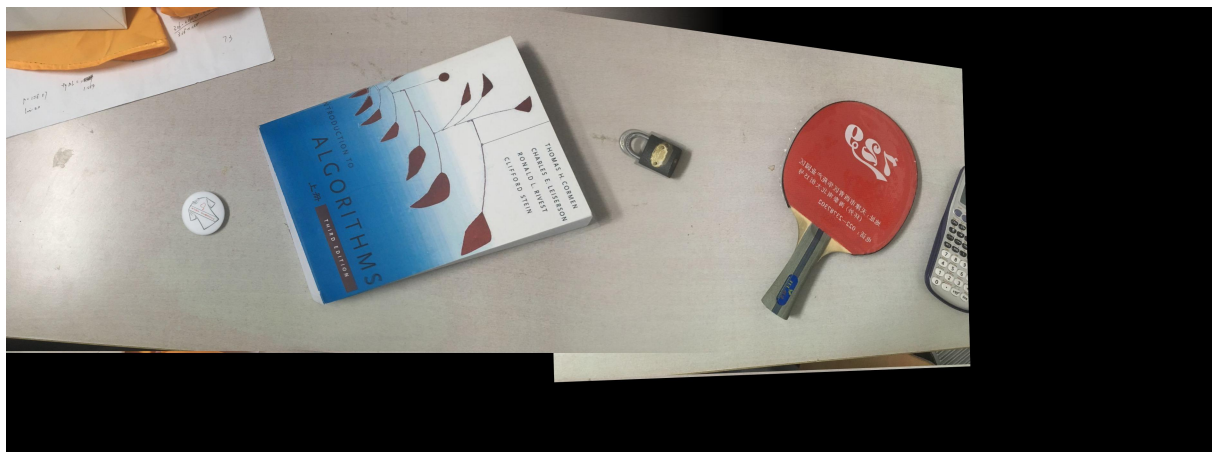
通过观察可以看到，匹配点之间没有误匹配发生，结果较好。

(4) 利用 **RANSAC** 提纯并求出单应矩阵。这里使用 `opencv` 中的 `findHomography()` 函数对两幅图中的匹配点进行提纯并求出单应矩阵。其中 **RANSAC** 的置信度为默认的 **0.99**。

(5) 对两幅图像进行拼接。对于单独的区域只取对应得图像即可，对于重叠区域采用加权求和的方法。其中权重为像素点距离边界线的距离占的比重。

五、实验结果

最终图像：



原始图像：



六、实验总结

通过本次实验，我了解了 SIFT 算法的原理，并了解了图象拼接的大致流程。由于这是本人第一次使用 opencv，所以中间遇到了很多坑。首先是 opencv 中坐标的定义问题。Opencv

中默认 x 轴为横轴， y 为纵轴，左上角为原点。这与矩阵的下标定义刚好相反。同时 `opencv` 中定义了非常多的类，但是官方文档写的非常详细，`demo` 也非常 `nice`。通过阅读官方文档能够快速学会使用 `opencv`，同时在完成实验的过程中，也参考了很多网上的博文。感觉收获良多！