

复旦大学 计算机学院

2018-2019 学年 第 1 学期 期末课程论文

课程名称: 计算机视觉 课程代码: **COMP130124.01**

开课院系: 计算机学院 考试形式: 课程论文

姓名：_____ 本人签名：_____

学号: _____ 专业: _____

提示：请同学们秉持诚实守信宗旨，谨守考试纪律，摒弃考试作弊。学生如有违反学校考试纪律的行为，学校将按《复旦大学学生纪律处分条例》规定予以严肃处理。

部分	1	2	3	4	5	6	7	8	总分
得分									

计算机视觉实验报告

熊倩 15307130228

项目主题

图片特征点匹配

课题介绍

图片特征点匹配，是计算机视觉任务中关键点检测中的一种，旨在通过对给定两张图片中的特征点进行匹配，来判断给定的两张图片是否有相似的特征，从而实现寻找特定物品，人像和场景的应用。图片特征点匹配算法的过程主要有四个步骤：获取特征点，特征点匹配，挑选匹配结果，描绘匹配点。

下面我会依次介绍本次实验中，上述四个步骤的具体实现过程及相关原理，然后我会将期中与本次实验所选择的不同方法进行对比，分析各自的优缺点。

实验过程

1. 获取特征点

在对高维图像数据进行分析的过程中，我们往往希望得到对图像精确而简短的表达，从而既能够有效地表达图像中的信息，也不会产生太大的计算量。图像特征就是对图像精确而简短的表达的一种方式，所以我们可以提取出图像中的关键信息，一些基本元件以及它们的关系，然后再对这些信息进行分析和研究。

在本实验中，我们首先需要获取图像中的特征点，然后再对特征点进行匹配。特征点提取主要有 SIFT，SURF 以及 ORB 三种方法，这三种方法在 OpenCV 里面都已得到实现。在第一次特征点匹配的实验中，使用了 ORB 方法，而本次实验使用的是 SURF 方法。下面我将简单介绍一下这三种方法的来源，过程和特点。

SIFT 算法：

SIFT(Scale-invariant feature transform)中文名为尺度不变特征转换，是一种计算机视觉算法，用来侦测与描述影像中的局部性特征，它在空间尺度中寻找极值点，并提取出其位置、尺度、旋转不变量，此算法的作者是 David Lowe。

SIFT 算法的实质是在不同的尺度空间上查找特征点，并计算出特征点的方向。SIFT 所查找到的特征点是一些十分突出，不会因光照，仿射变换和噪音等因素而变化的点，如角点、边缘点、暗区的亮点及亮区的暗点等。

David Lowe 将 SIFT 算法分解为如下四步：

1. 尺度空间极值检测：搜索所有尺度上的图像位置，通过高斯微分函数来识别潜在的对于尺度和旋转不变的兴趣点。

2. 关键点定位：在每个候选的位置上，通过一个拟合精细的模型来确定位置和尺度。关键点的选择依据于它们的稳定程度。

3. 方向确定：基于图像局部的梯度方向，分配给每个关键点位置一个或多个方向。所有后面的对图像数据的操作都相对于关键点的方向、尺度和位置进行变换，从而提供对于这些变换的不变性。

4. 关键点描述：在每个关键点周围的邻域内，在选定的尺度上测量图像局部的梯度。这些梯度被变换成一种表示，这种表示允许比较大的局部形状的变形和光照变化。

SIFT 算法具有尺度不变形，独特性好，多量性，高速性和可扩展性等优点，较好地克服了图像的旋转、缩放、平移，图像仿射（投影变换）、光照、遮挡、杂物场景和噪声等干扰。

SURF 算法：

SURF 算法（Speeded-Up Robust Features），中文名为加速鲁棒特征。2006 年，Bay 和 Ess 等人在 SIFT 算法的基础上进行改进，提出了 SURF 算法。该算法在保持 SIFT 算子优良性能特点的基础上，解决了 SIFT 计算复杂度高、耗时长缺点，对兴趣点提取及其特征向量描述方面进行了改进，计算速度得到了提高。SURF 算法的关键步骤有构造 Hessian 矩阵，计算特征值 α ，构造高斯金字塔，定位特征点，确定特征点主方向，构造特征描述子等。

ORB 算法：

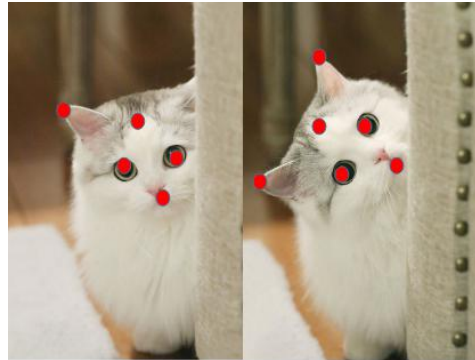
ORB 算法（Oriented FAST and Rotated BRIEF），是对 FAST 特征点与 BRIEF 特征描述符的一种结合与改进。FAST 是一种特征检测算法，而 BRIEF 是一个特征描述符，是图像一种表示方式，可以通过比较两个图像的关键点描述符，进而判断关键点是否匹配。ORB 使用 oFAST 进行特征点检测，并在提取的特征点加上一个特征点方向，然后使用带方向的 rBRIEF 算法高效的对特征点描述符进行计算，并在点对的挑选上使用贪婪搜索算法，挑出了一些区分性强的点对用来描述二进制串。

通过上述三种算法，我们可以检测出图片中的特征点，其结果示意如下：

原图：



检测出的特征点：



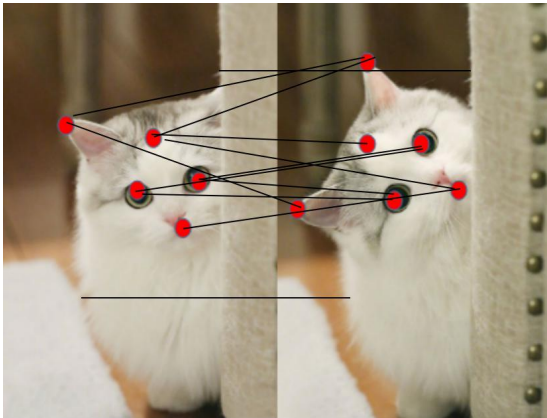
2.特征点匹配

在使用 SIFT, SURF 或者 ORB 算法找出特征点或者关键点之后,我们需要对两张图片中的特征点进行匹配,进而观察特征点匹配的程度如何。

OpenCV 提供了两种特征点匹配的方式: Brute-force matcher (cv::BFMatcher)和 Flann-based matcher (cv::FlannBasedMatcher)。Brute-force matcher 就是对点集一中每个特征点描述符,遍历点集二,并找到之距离最近的特征点描述符;而 Flann-based matcher 使用快速近似最近邻搜索算法寻找,相对暴力匹配来讲更加快速和使用方便。FLANN 具有一种内部机制,可以根据数据本身选择最合适的算法来处理数据集,并且 FLANN 匹配器只能使用 SURF 和 SIFT 算法。

在第一次实验中使用的是暴力匹配,而本次实验使用的是 FLANN 匹配器。

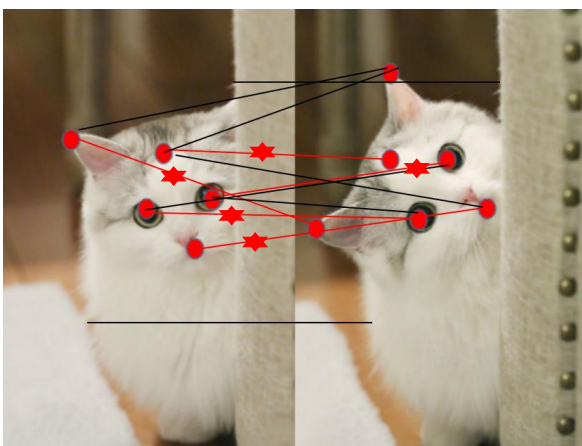
对上述两张猫的图片的特征点匹配结果示意图如下:



3.挑选匹配结果

在步骤二中得到的匹配结果有好有坏,所以我们需要在所有匹配结果中挑选出较好的匹配结果。由于特征点对的匹配距离越近,则匹配结果越好,所以我们就选择那些匹配距离更近的特征点对。

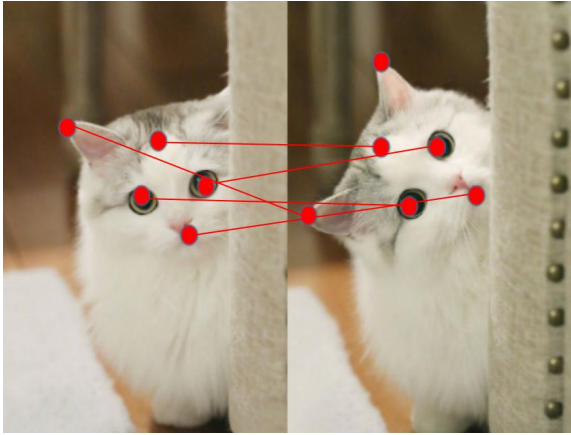
挑选之后的特征点匹配结果示意图如下,线条上打星号的特征点匹配即是挑选之后的匹配结果。



4.描绘匹配点

最后,我们只需要去除剩余的匹配,把挑选得到的结果描绘出来即可。

描绘结果示意图如下:

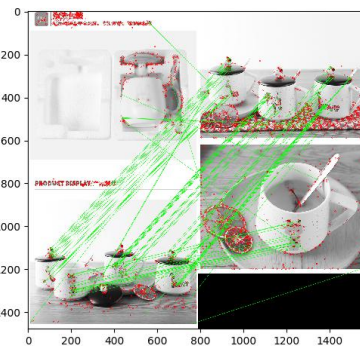


两次实验结果对比

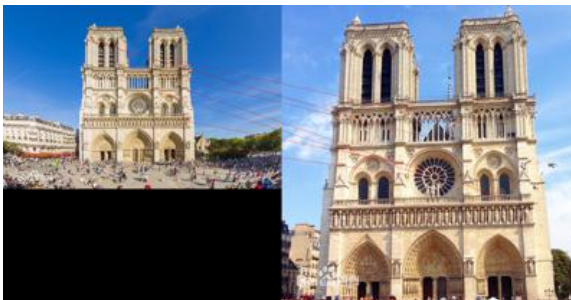
第一次实验的结果在左边，本次试验的结果在右边，对比如下：



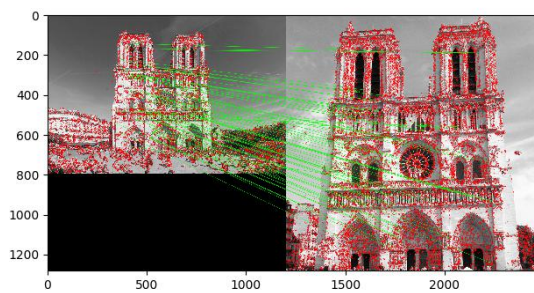
第一次实验



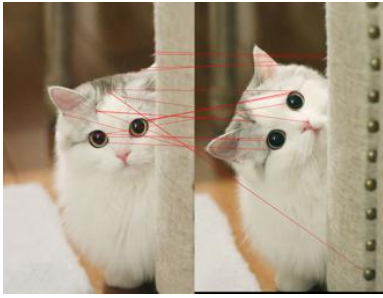
第二次实验



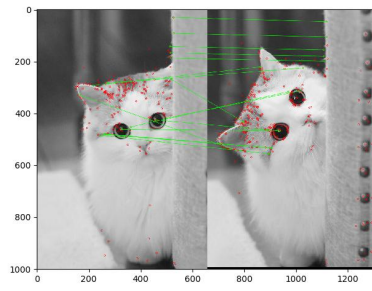
第一次实验



第二次实验



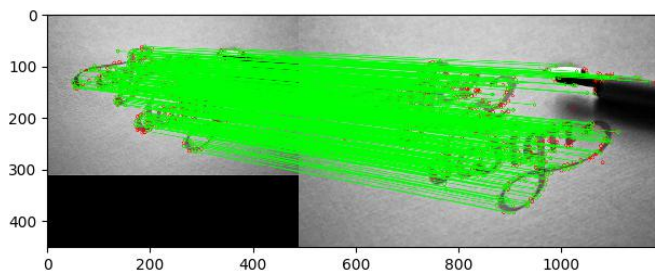
第一次实验



第二次实验



第一次实验



第二次实验

从上述四处图片对比中，我们可以看到，两次实验都能初步检测出图片中的特征点和找到特征点间的匹配，也都会把图片中的一些噪点当成特征点。但第二次实验的改进之处在于，抗干扰能力更强，较少把噪点识别为特征点，并且匹配时的全局感更好，考虑了整张图片的信息。

方法比较与总结

图片特征点匹配算法共有四个步骤。两次实验在最后两个步骤，即挑选匹配结果和描绘匹配点的方法上并没有太大的差异；而在前两个步骤中，即获取特征点和特征点匹配上是采用了不同的方法的。第一次实验在获取特征点时采用了 ORB 算法，在特征点匹配上采用了暴力匹配；本次实验在获取特征点时采用了 SURF 算法，在特征点匹配上采用的是 FLANN 匹配器。下面就对获取特征点和特征点匹配两个步骤中的不同方法进行比较和总结。

1. 获取特征点：

第一次实验使用了 ORB 算法，该算法基于 FAST 特征点与 BREIF 特征描述符，并不在 SIFT 和 SURF 算法的框架中。ORB 算法的优点是，速度比 SIFT 和 SURF 算法的速度都要快，但是 ORB 算法的缺点是，它没有尺度不变的特征，在旋转鲁棒性，模糊鲁棒性和尺度变换鲁棒性上没有 SURF 算法好。

第二次实验使用了 SURF 算法，该算法是在 SIFT 算法的基础上进行改进的，保留了 SIFT 算法尺度不变的特性，并且计算速度得到了提高。所以第二次实验的优点是抗干扰能力相比于实验一更强，较少把噪点识别为特征点，但缺点是速度变慢了。

因此，如果对速度要求较高，可以选用 ORB 算法，但要确保图片较为规范；如果对适用性要求较高，可以使用 SURF 算法。

2.特征点匹配：

第一次实验使用了暴力匹配器，它的优点是总是尝试所有可能的匹配，从而使得它总能够找到最佳匹配，也就是找到点集二中距离点集一中某点最近的点。它的缺点是由于其尝试所有可能的匹配，计算时间较长，并且需要对计算出来的特征点进行抗噪处理。

第二次实验使用了 FLANN 匹配器，它的优点是速度更快，但缺点是它找到的是最近邻近似匹配，不能确保找到最佳匹配。

所以当我们需找出最佳匹配时，我们可以使用暴力匹配器，然后对匹配结果进行细致地处理；当我们只需要找到一个相对较好的匹配时，往往使用 FLANN 匹配器。

3.下一步的改进：

这两次实验只是使用了 OpenCV 里面现有的方法进行初步的尝试，我们还可以进一步对实验的步骤进行改进，对现有的寻找特征点方法和匹配器进行改进和调整，也可以寻找其他的方法来代替上述提及的方法，也可以尝试搭建新的方法。

参考文献

- [1]David G.Lowe Distinctive Image Features from Scale-Invariant Keypoints. January 5, 2004.
- [2]Ethan Rublee, Vincent Rabaud, Kurt Konolige, Gary R. Bradski: ORB: An efficient alternative to SIFT or SURF. ICCV 2011: 2564-2571.
- [3]<https://blog.csdn.net/zilanpotou182/article/details/66478915> SURF SIFT ORB 三种特征检测算法比较
- [4]<http://www.cnblogs.com/ronny/p/4260167.html> 图像局部特征点检测算法综述
- [5]<https://blog.csdn.net/zddbblog/article/details/7521424> SIF 算法详解
- [6]<https://blog.csdn.net/lhanchao/article/details/52612954> 特征点匹配——ORB 算法介绍
- [7]https://blog.csdn.net/HuangZhang_123/article/details/80660688 Python 使用 Opencv 实现图像特征检测与匹配
- [8]<https://blog.csdn.net/cshilin/article/details/52107580> 高维数据的快速最近邻算法 FLANN
- [9]https://blog.csdn.net/fb_help/article/details/70048356 Brute Force 匹配（暴力匹配）和 FLANN 匹配区别
- [10]https://blog.csdn.net/GAN_player/article/details/78285771 特征匹配之 Brute-Force 匹配和 FLANN 匹配器