

《视觉定位与感知》课程

期末大作业报告

实验名称: 相机位姿估计问题

姓 名： 陈烨 \_

学 号： 516030910254 \_\_\_

班 级： F1603402 \_\_\_

2018年 1 月 11 日

# 问题描述

通过本课程的学习，我们知道通过若干张图像，我们可以同时获得图像中所拍摄场景的三维信息，并得到图像拍摄时的6个自由度的姿态和位置。

本课程大作业问题内容是使用自己的手机或相机，对同一个场景，分别在三个角度拍摄三张图像，最终获得三张图像拍摄时的姿态和位置，以及图像中特征点的三维坐标信息，并评估和验证所实现算法的正确性和精度性能。在本次实验中，我使用iphone手机拍摄的照片为交大“庙门”。三张图片如图1-3所示：



图1. 拍摄第一张图片



图2. 拍摄第二张图片



图3. 拍摄第三张图片

注：在拍摄时，所选择的场景纹理丰富，并且三维结构突出；三个拍摄角度间既要有足够的基线，也要控制相互夹角不大于30度，不然会造成后续特征点匹配的困难。

实验过程为利用拍摄的三张图片估计三次拍照的相对位姿。在实验过程中以第一次拍照的位置为轴，用相对旋转矩阵与平移矩阵来表示另外两次拍照的位姿。

# 实验原理

* 1. 原理简述

首先选择其中两张照片，通过特征点匹配基础矩阵或本征矩阵的求解得到两个拍摄视角的相对位姿，然后使用三角化得到对应特征点的三维点云。对于第三张图，我们可通过相机位姿估算的方式求解其姿态和位置。在这里用的方法是通过第一张图片与第三张图片的特征点与三角化的三维点云的3D-2D匹配关系，利用RANSAC方法实现第三个视角的相机位置姿态估算。

* 1. **实验原理具体阐述**
     1. **相机内参标定**

通过MATLAB工具箱进行相机内参标定，需要用同一个相机对某个图案（这里选择黑白棋盘）拍摄多个位姿的图片，按照官方示例进行标定。其中拍摄的20张黑白棋盘的位姿图片如图4所示：

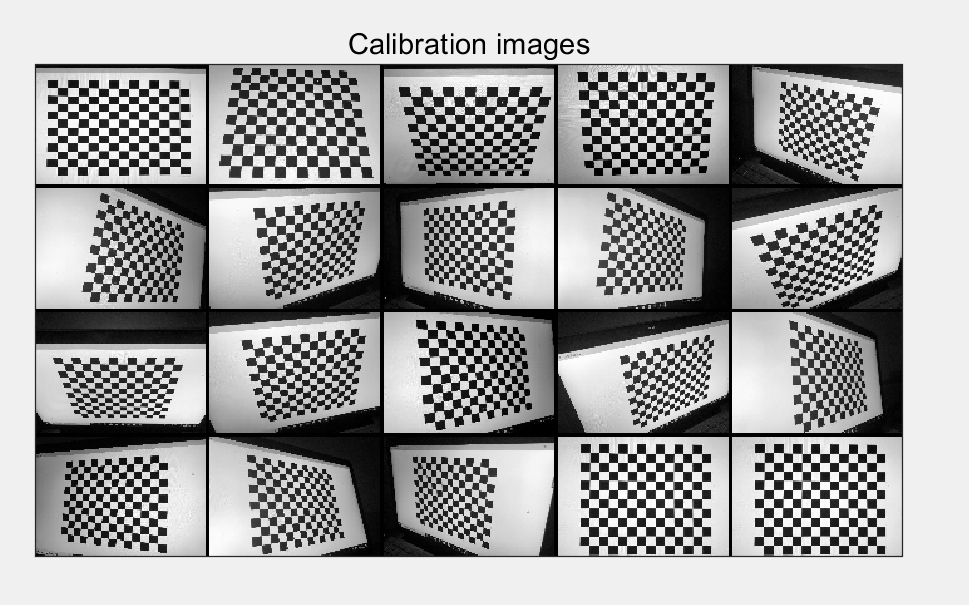


图4. 用于相机内参标定的20张图片

* + 1. **去畸变原理**

对于想要的无畸变的目标图像，从已畸变的图像找出对应的像素，将该像素颜色填入目标图像，以此构建像素映射完成去畸变图像。

* + 1. **SURF特征匹配原理**

特征点的匹配是通过计算两组特征点的128维的特征点的欧氏距离实现的。欧氏距离越小，则相似度越高，当欧式距离小于设定阈值时，可以判定为匹配成功。

Sift算法的优点是特征稳定，对旋转、尺度变换、亮度保持不变性，对视角变换、噪声也有一定程度的稳定性；缺点是实时性不高，并且对于边缘光滑目标的特征点提取能力较弱。Surf改进了特征的提取和描述方式，用一种更为高效的方式完成特征的提取和描述，具体实现流程如下：

1. 构建Hessian（黑塞矩阵），生成所有的兴趣点，用于特征的提取；

2. 构建尺度空间

3. 特征点定位

4. 特征点主方向分配

5. 生成特征点描述子

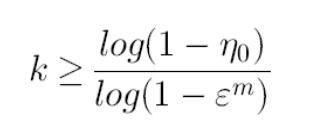
6. 特征点匹配

* + 1. **利用RANSAC进行基础矩阵与本征矩阵求解原理**

在经典的RANSAC流程中，目标函数**C**可以被看作：在第k次迭代过程中，在当前变换参数IMG_256作用下，数据集IMG_257中满足变换参数的点的个数，也就是在当前变换条件下类内点的个数，而RANSAC就是最大化 **C**的的过程。而判断当前某个点是否为类内需要一个阈值**t。**

 在迭代的过程中，当前变换参数 θ 的计算需要IMG_256中的一个子集 **I**来计算，RANSAC是一个随机从IMG_257中采样一个子集，然后对参数“估计-确认”的循环。每一个子集应是一个大小为m 的最小采样。所谓最小采样，就是**m**的大小刚好满足计算**θ**的个数即可。

在置信度为IMG_256的条件下，在循环过程中，至少有一次采样，使得采样出的**m**个点均为类内点，这样才能保证在循环的过程中，至少有一次采样能取得目标函数的最大值。因此，采样次数k应该满足以下条件：

 （1）

这里除了置信度IMG_256外，**m**为子集大小，ε 为类内点在IMG_257中的比例，其中置信度一般设置为[0.95, 0.99]的范围内。然而在一般情况下，**ε** 显然是未知的，因此 **ε** 可以取最坏条件下类内点的比例，或者在初始状态下设置为最坏条件下的比例，然后随着迭代次数，不断更新为当前最大的类内点比例。

* + 1. **三角化进行点云重构原理**

如图5所示，三角化的过程即为在已知两个成像平面中对应的点位置（）的情况下，如果知道两个成像视角的相机矩阵（），就可以重构出该对应点在三维空间中的位置。

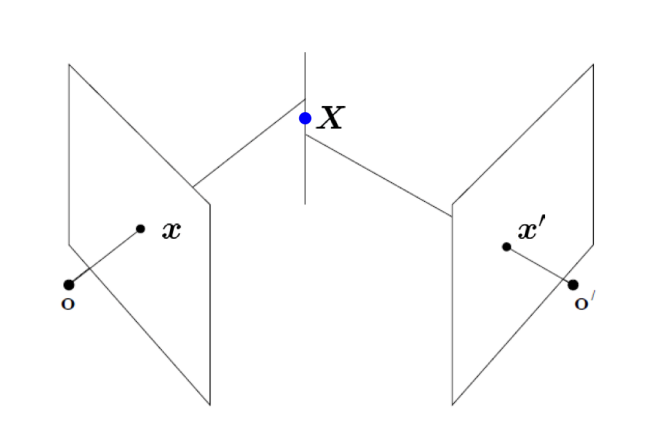
****

图5. 三角化进行三维点云重构

# 算法实现思路

该算法实现共分9个步骤：

1）标定相机的内参；

2）选择一个合适场景，用已标定的相机拍摄三张图像；

3）去除畸变影响；

4）选取其中两张图进行特征匹配；

5）使用RANSAC方法进行基础矩阵和本征矩阵的估算；

6）从基础矩阵或本征矩阵中解出对应的位姿变换（旋转矩阵与平移矩阵）；

7）利用三角化进行三维点云重构；

8）将第一张图片与第三张进行特征匹配，并利用已经三角化的特征点得到第三张图的3D-2D对应关系；

9）再次使用RANSAC方法估计第三个相机位置。

在本次实验中利用MATLAB工具箱实现了以上9个步骤。

* 1. **相机内参标定实现**

利用MATLAB的“TOOLBOX\_calib”工具包，参考 *<http://www.vision.caltech.edu/bouguetj/calib_doc/>*

进行标定。其中我使用的二十张标定图片如图4所示，标定得到的相机参数如图6所示：

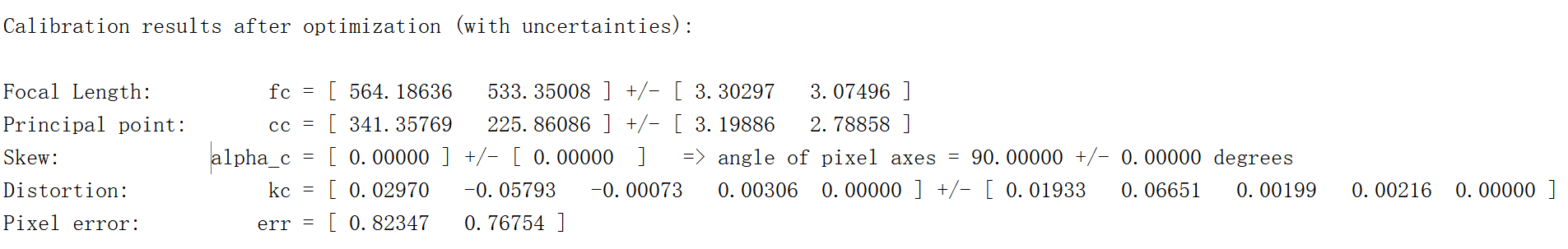


图6 相机内参描述

* 1. **去畸变实现**

MATLAB自带去除畸变的函数unditorImage，输入相机内参以及由该相机拍摄的图片即可获得去畸变处理后的图像。函数使用示例如图7所示：



图7.去畸变函数使用

* 1. **拍摄三张图片**

拍摄图片及要求如图1-3所示。

* 1. **对两张图进行特征点匹配**

在实现特征点匹配的算法中，使用SURF描述子。利用MATLAB的Computer Vision System Toolbox工作箱，对两张图片首先分别利用detectSURFFeatures函数检测图像的SURF特征点，该函数输入为需要提取特征的图像，返回提取得到的SURF特征点。然后利用extractFeatures函数对上一步提取得到的特征点计算描述向量用于之后的匹配。extractFeatures函数的输入为图片以及对应的特征点，返回值为SURF描述子（特征描述向量）以及对应该描述子的点的位置。在得到了两个图像的SURF描述子之后就可以用该描述子进行点的匹配，利用描述子做匹配的过程主要运用了matchFeatures函数，该函数输入为两个图像的特征描述子，返回的是匹配之后的匹配点的索引，index，其中index为n\*2的矩阵，第一列对应第一张图像的匹配点的索引，第二列对应第二张图像的匹配点的索引。有了该索引，结合extractFeatures函数输出的描述向量对应点的位置，即可得到两张图片的匹配点。在匹配的过程中，为了降低outlier的比例，利用最佳匹配和第二好匹配的特征描述子距离比值进行筛选，对应于matchFeatures函数的’Maxratio’参数，在这里按照老师的建议设置为0.7。在得到了匹配点的位置之后就可以进行显示，显示函数为showMatchedFeatures，该函数的输入为两张图像及对应的匹配点在图像中的位置。用MATLAB的Computer Vision System Toolbox工作箱实现上述匹配过程的示例如图8所示：

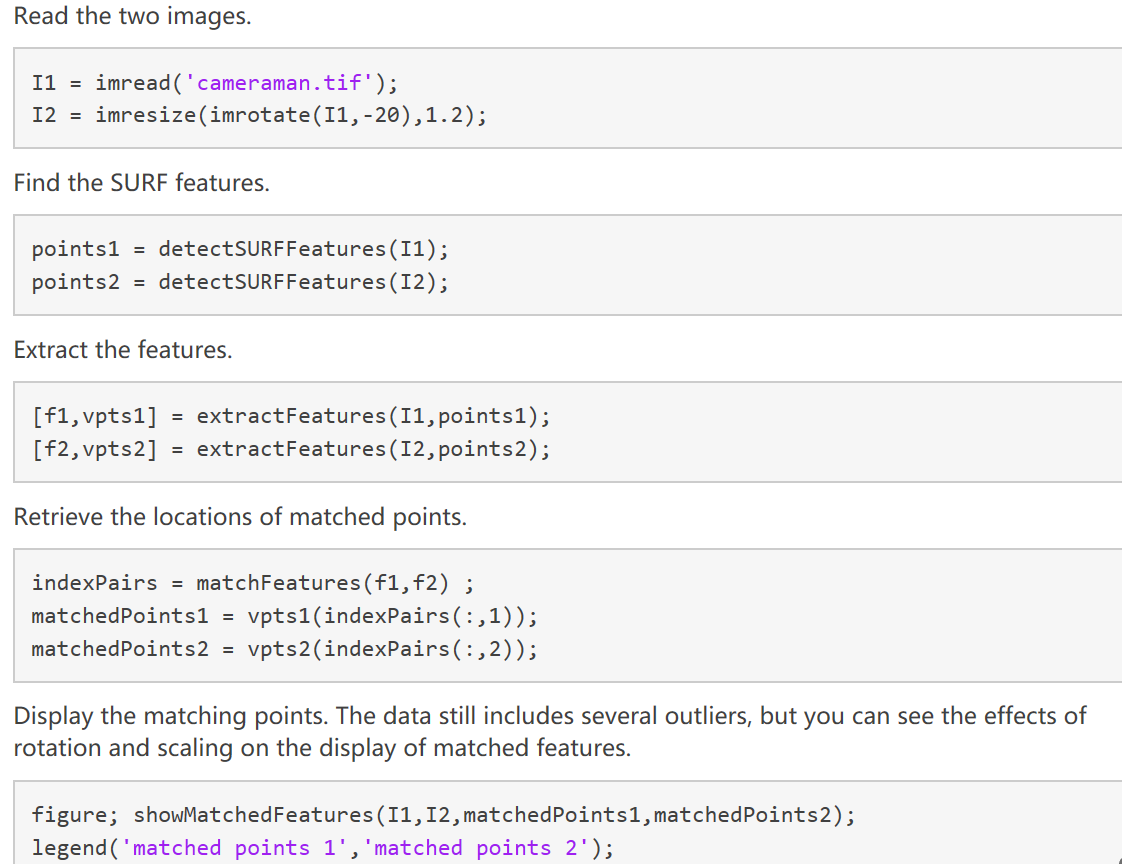
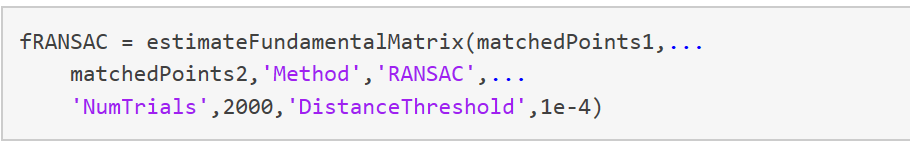


图8. 用MATLAB的Computer Vision System Toolbox工作箱实现 SURF描述子匹配过程

* 1. **使用RANSAC进行基础矩阵的估算**

利用RANSAC进行基础矩阵及本征矩阵的估算的实现，可以调用[Computer Vision System Toolbox](https://ww2.mathworks.cn/help/vision/index.html) 工具箱中的estimateFundamentalMatrix和estimateEssentialMatrix函数。在这里我实现的思路是先用第4步求出的匹配点集，调用estimateFundamentalMatrix函数，该函数的输入为两组相互匹配的点集，以及一些（name，value）对，在这里设置’Method’为‘RANSAC’,即使用RANSAC方法对该匹配进行基础矩阵求解。该函数在返回基础矩阵F的同时还返回了对应于该基础矩阵的内点。在这个迭代过程之后，将返回的内点送入estimateEssentialMatrix函数再计算一遍本征矩阵，由于输入已经是内点，且考虑时间成本，该计算过程不需要RANSAC迭代，可直接得到对于本征矩阵E的较精确的估计。基础矩阵与本征矩阵的估计示例如图9所示：



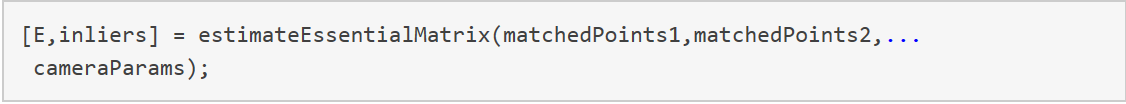


图9 利用MATLAB工具箱进行基础矩阵与本征矩阵的估计示例

* 1. **从基础矩阵中分解旋转矩阵与平移矩阵**

显然在通过两个图像点的匹配求得对应的本征矩阵之后，可直接由本征矩阵结合相机参数估算两个视角的相对旋转矩阵与平移矩阵。该过程可利用MATLAB中的relativeCameraPose函数和cameraPoseToExtrinsics函数。其中relativeCameraPose函数输入为本征矩阵、相机内参以及对应的匹配内点。该函数的作用即将联系两张图像的本征矩阵转化成对应的相机相对位置。返回的参数即相对方向和相对距离。将这两个返回参数送入cameraPoseToExtrinsics函数就可以将该相对方向和距离转化为相对旋转矩阵和平移矩阵。对于第一张图像的相机位姿而言，它在这个过程中是起到参照物的作用，因此它的相对方向和距离矩阵分别为单位阵和零矩阵。将单位阵和零矩阵输入cameraPoseToExtrinsics函数便可得到在第一个相机位姿的参照下，其本身的旋转矩阵和平移矩阵。由本征矩阵求出旋转矩阵和平移矩阵的方法示例如图10所示：





图10. 由本征矩阵求出旋转矩阵和平移矩阵的方法示例

* 1. **三角化三维点云重构**

如2.2.5中的原理所述，当我们知道两张图像的对应匹配点，同时还得到了两个成像相机的相机矩阵，就可以用三角法重构出这些点在三维坐标下的位置。在MATLAB实现的过程中，调用了cameraMatrix函数，输入相机参数以及对应的相机的旋转矩阵和平移矩阵，返回该相机矩阵。对两个相机位姿拍下的图分别调用该函数，即可得到两个相机矩阵，将相机矩阵与两张图中的匹配点输入triangulate函数进行三角化，即可得到在三维坐标系下各点的位置。三角化三维点云重构示例如图11所示：

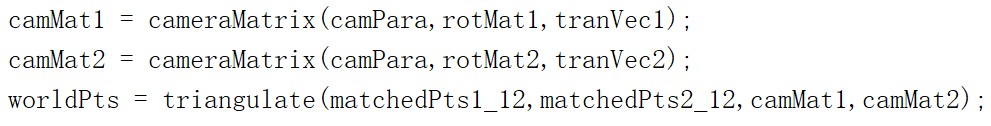


图11.三角化三维点云重构

* 1. **得到第三张图的3D-2D对应关系**

我们以及得到在第一个相机参照下的三维点云，我们再对第一张图片与第三张进行SURF特征匹配，并根据匹配点在三维坐标系下的坐标得到对应第三个相机的部分点在三维坐标系下的坐标，由于我们的三维坐标系是以第一个相机为中心建立的，因此由这些点在三维坐标系下的位置我们又可以估算第三个相机相对第一个相机的位姿。

在MATLAB实现时仍然参照3.4的方法进行图片1和图片3的匹配，求得对应匹配点的索引，利用该索引找出三维坐标系中的点即可，这些点即第三张图像的点在三维坐标系下的映射。由此便构建了2D-3D对应关系。

* 1. **使用RANSAC方法估算第三个视角位置**

如3.8所述，我们已经知道了第三张图片的点在三维空间中的坐标，我们就可以利用MATLAB的estimateWorldCameraPose函数估算出在三维空间里的相机3的方向与距离矢量。由于我们的三维空间是以第一个相机的位姿为中心建立的，因此这个位姿就是相机3相对相机1的位姿（对应P3P算法）。在MATLAB实现过程中首先用estimateWorldCameraPose函数估算相机3在三维空间里的方向和距离，再利用cameraPoseToExtrinsics函数得到该方向和距离对应的旋转矩阵和平移矩阵，该矩阵就是在相机1的参照下的相对位置。

# 实验结果与分析评估（定性+定量）

# 实验结果的验证主要分定性和定量两个环节。其中定性分析就是把三维点云与3个相机位姿标出在三维坐标系中，观察是否与拍照时的情况相匹配。

其中，通过SURF描述子初步匹配的情况如图12所示，经过RANSAC求解本征矩阵后的内点匹配情况如图13所示，转换到三维坐标系下的点与相机位置如图14所示，其中三个”星号“为相机位置。

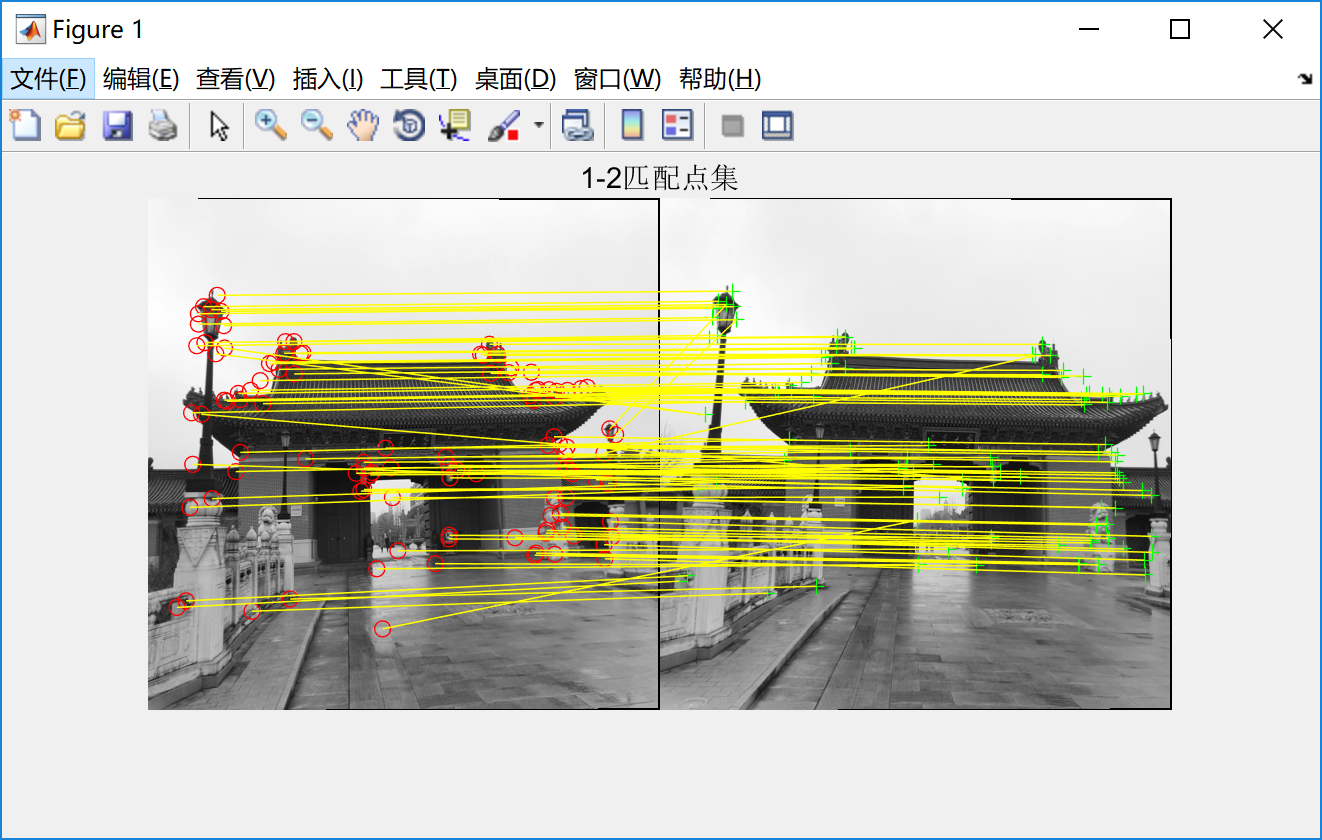


图12. SURF描述子初步匹配的情况

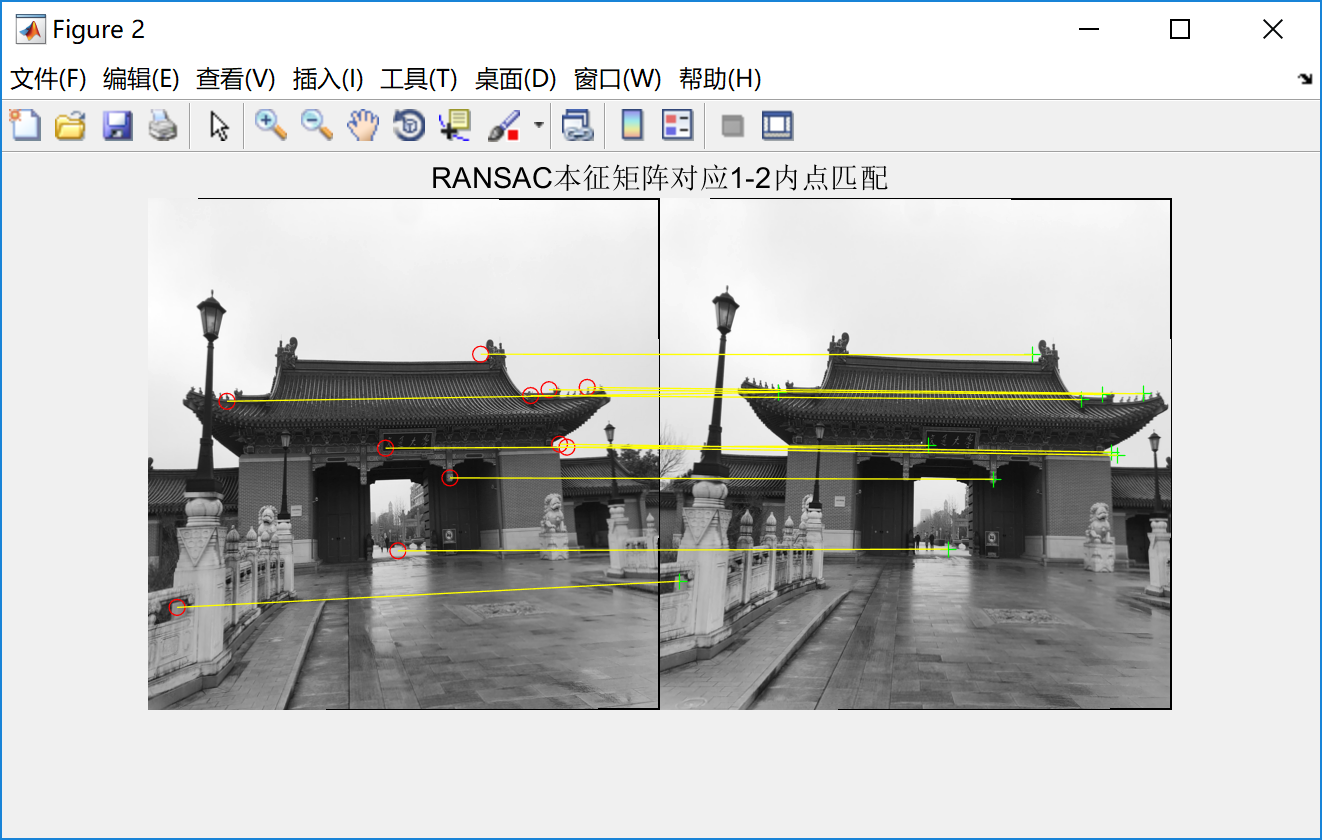


图13. 求解本征矩阵后的内点匹配情况

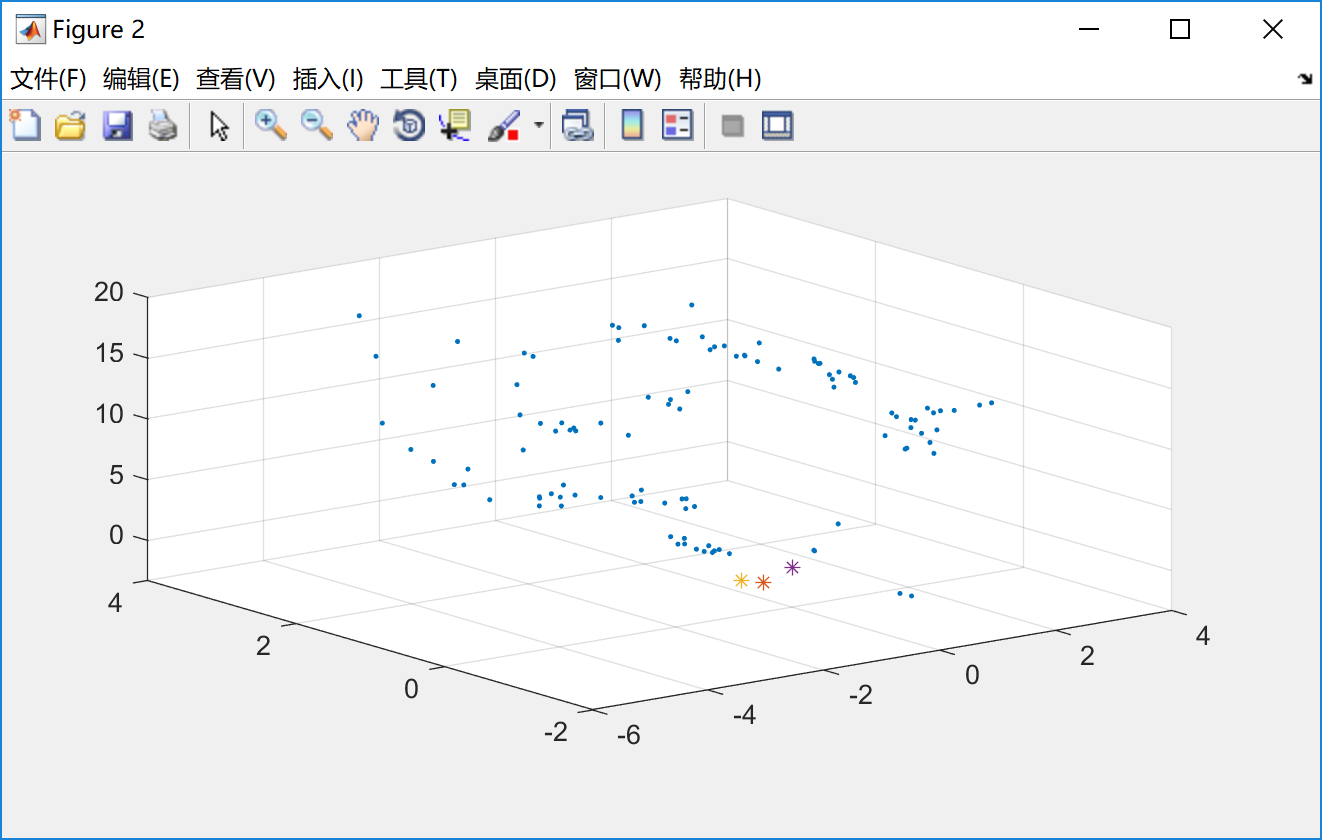


图14 相机位姿与图像主要特征点的位置关系

分析：可以看到，将特征点与相机位置放到三维坐标系下，可以定性地看出拍照时的实际站位关系，因此算法基本正确。

至于定量的分析评估，我这样考虑：我用两种方法求解，法一即第一张图片与第二张图片进行匹配可以估计第二个相机的相对位姿，再进行三角化三维点云重建，再构建第三张图片的2D-3D关系从而得到第三个相机相对第一个相机的位姿，法二是直接拿第一张图片与第三张图片进行匹配，则可以直接迭代求解本征矩阵进而分解出第三张图片的相对旋转矩阵与平移矩阵。上述两种方法求得的旋转矩阵理论上应该是相等的，因此定量比较这两个旋转矩阵即可评估算法正确性。同时，对于三张图片我们可以进行1-2匹配估计3、1-3匹配估计2、2-3匹配估计1三种评估。定量评估结果如图15-17所示：

其中：M\_rot3\_1\_new为法二求得的待估计的第三个相机的位姿（在三种情况里分别对应相机3、2、1）；

M\_rot3\_1为法一求得的第三相机的位姿。

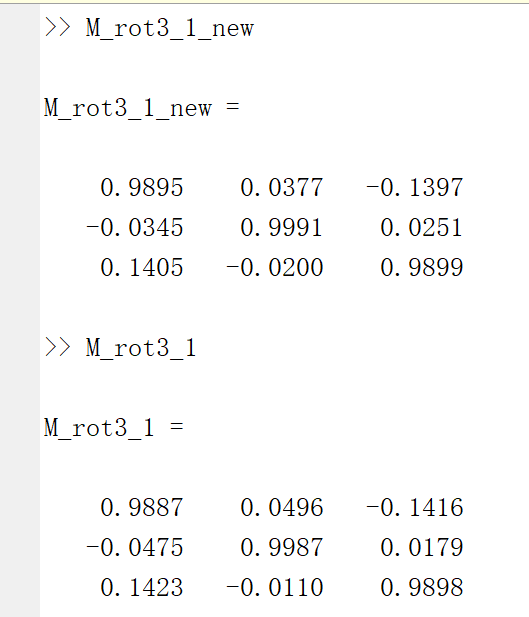


图15. 由1-2匹配求解3的相机位姿的旋转矩阵

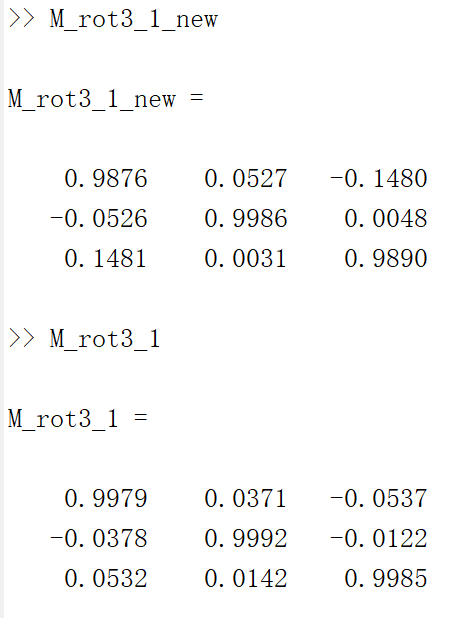


图16. 由1-3匹配求解2的相机位姿的旋转矩阵

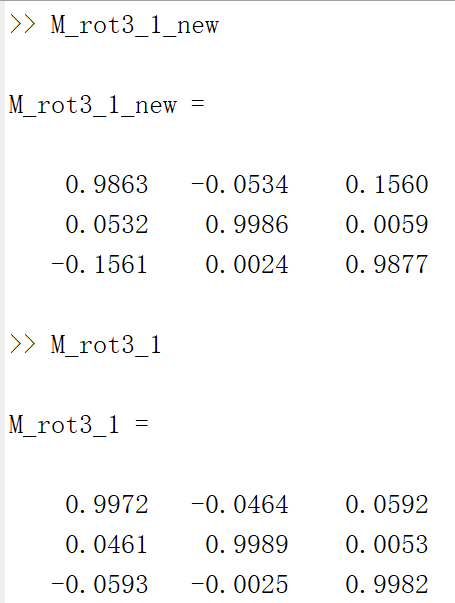


图17. 由1-3匹配求解2的相机位姿的旋转矩阵

分析：由图15-17可以看出，无论怎样安排匹配图片，用两种方法最终求得的第三个相机的位姿的旋转矩阵，每个元素的误差都在0.01左右或以内，因此可以定量地验证本次实验算法地误差。可以认为算法正确。

# 实验总结

这门课是我非常感兴趣地计算机视觉方面的课，非常感想邹老师与助教老师的辛勤教导与答疑。我自认为对这门课非常有热情，因此即算是课堂内容对我来说比较高阶，在课堂上很难理解消化，作业也偏难，但也从未放弃对相关知识的追求。本次期末大作业由于时间原因，是在考试周边复习考试边完成的，因此有些算法调用了MATLAB的工具箱，还未有时间自己去实现，但在假期里一定不遗余力地继续完善。再次真诚地感谢老师地指导，助教地答疑。