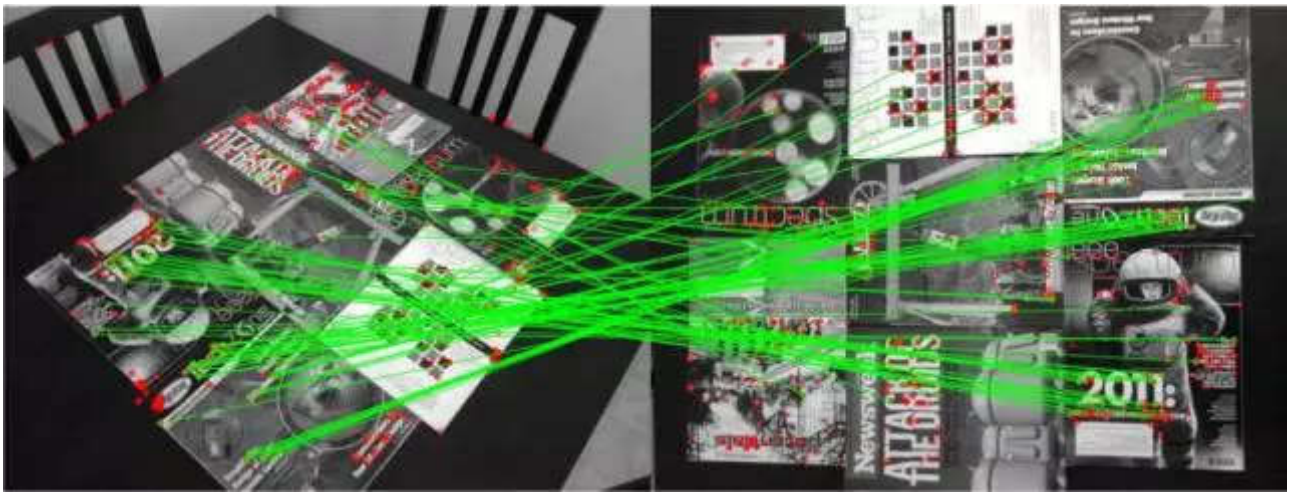
视觉里程计：特征点法之全面梳理



前一篇文章《视觉里程计：起源、优势、对比、应用》里介绍了视觉里程计（visual odometry，简称VO）的发展起源、技术优势、和SFM/SLAM的对比以及应用场景。本文介绍VO的具体技术。

根据VO是否需要提取特征点，VO的具体实现方法可以分为基于**特征点法的VO和直接法VO**（高翔《视觉SLAM十四讲》）。特征点法VO是以提取图像中的特征点为基础，学术界有长久的研究，运行比较稳定、对光照变化和动态场景鲁棒性强，是比较成熟的VO实现方案。而直接法VO是为了克服特征点法VO的部分缺点（如计算量大，不适用于纹理缺乏场景等）而出现的，随着一些直接法VO开源项目的出现，正逐渐进入主流。

本文首先介绍发展相对比较成熟的特征点法VO，下一篇将介绍直接法VO。

如何获得好的VO效果？

首先我们来讨论一下，VO想要获得好的结果，需要哪些条件?

1、**环境的光照强度比较合适，不能太暗或者太亮。**较暗的场景（如下图左）相机成像不清晰而且有很多噪点，而光线太强（如下图右）容易引起相机过度曝光，这些对于特征点提取非常不利。

左：光照不足，右：过度曝光

2、**环境具有相对丰富的纹理。**即使光照强度合适，VO的效果也和场景内容有关。具有丰富纹理的场景可以很容易提取出大量有效的特征点，而纹理单一的场景（比如极端的白墙等）缺乏可靠有效的特征点会使得VO性能大幅下降。如下图所示。

纹理丰富场景（左）和纹理单一场景（右）

3、**需要捕捉连续的图像，保证相邻图像帧的内容有足够的重叠区域。**因为需要对相邻图像重叠区域内的特征点进行匹配，所以重叠区域不能太小，否则可以匹配的特征点非常少，容易产生错误。

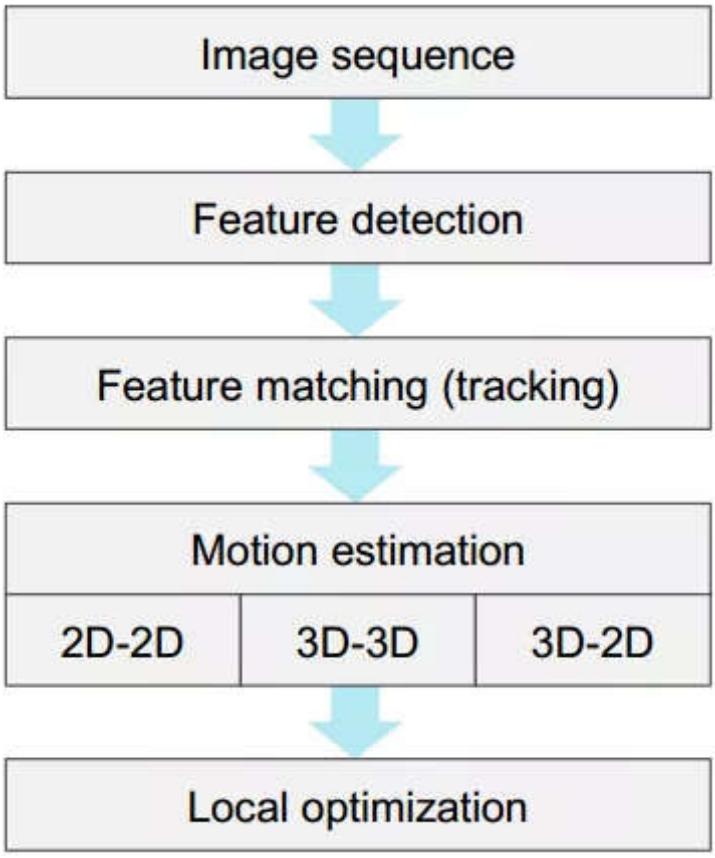
4、**最好是静态场景下运行。**而在有运动物体的动态场景下效果会变差。这是因为上述第3点需要在相邻帧间进行特征点匹配，如果场景中出现快速运动物体，比如人行走时

的手臂快速摆动，在相邻帧间相同的特征点位置是突变的，而静止物体的特征点位置是连续小幅变化的，这样在对特征点对进行一致性检验的时候会剔除突变的特征点对。

快速运动的人体很难做VO

特征点法VO算法梳理

特征点法VO的算法流程如下图所示：



特征点法VO流程 （Davide Scaramuzza and Friedrich Fraundorfe, 2011）

1 特征点检测

我们知道常见的彩色图像一般都是由RGB三个分量组成的，这种RGB格式的彩色图可以很方便的转换为灰度图。RGB和灰度的转换，实际上是人眼对于彩色的感觉到亮度感觉

的转换，这是一个心理学问题，其转换公式如下：

Grey = 0.299*R + 0.587*G + 0.114*B

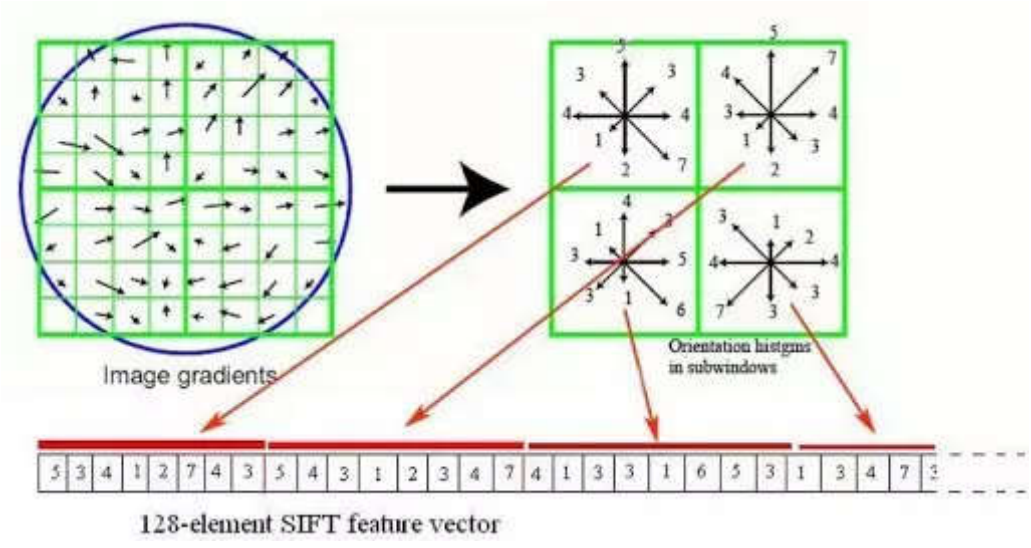
这样得到的灰度图就是由不同亮度值组成的矩阵，而**特征点直观的理解就是图像中（一般是灰度图）中那些具有代表性的特殊的点。**我们知道灰度值很容易受到光照、物体形变、图像拍摄角度等因素的影响，所以**特征点的选择和设计非常重要。**优秀的特征点应该具有以下特征：

- 1、**高辨识度。**能够和周围的像素进行明显的区分，具有极强的代表性。
- 2、**可重复性强。**相同的特征点可以在下一幅图像中被再次检测到，这样才能进行特征点匹配。
- 3、**高鲁棒性。**在光照变化、几何变化（旋转、缩放，透视变形）时能够保持没有明显变化，在图像中存在噪声，压缩损伤，模糊等情况下仍然能够检测到。
- 4、**计算效率高。**设计规则简单，占用内存少，有利于实时应用。但通常计算效率和鲁棒性成反比。

常见的比较知名的特征点有：

SIFT(Scale-invariant feature transform)

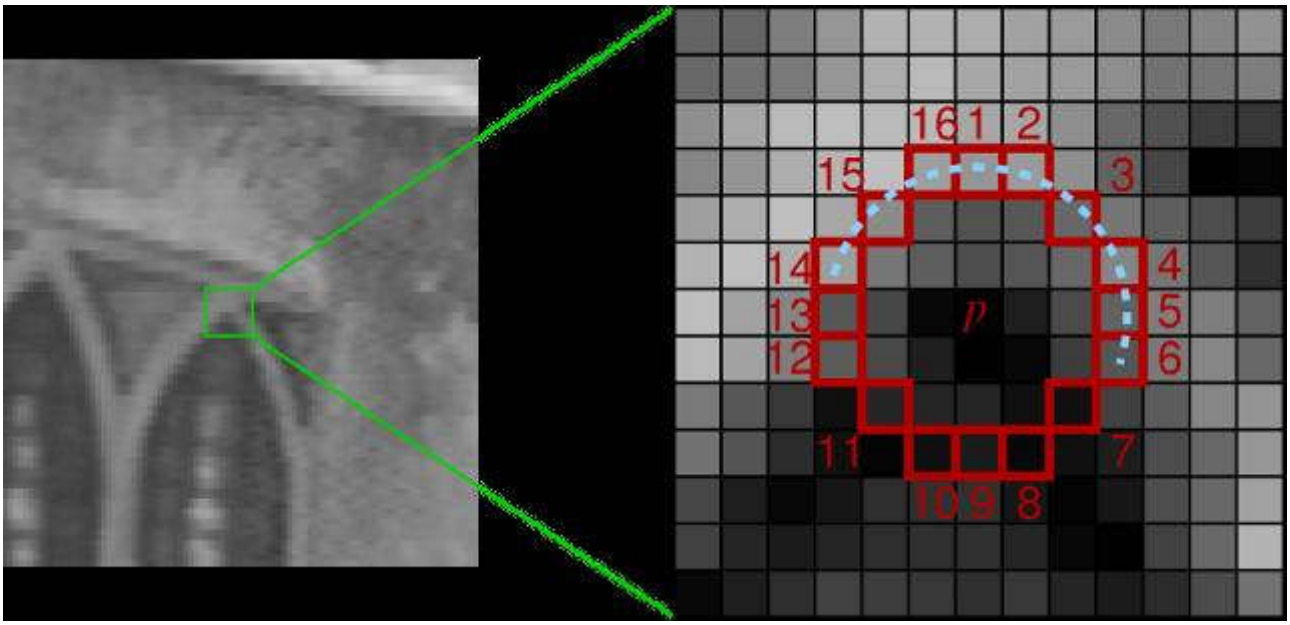
1999年提出，2004年完善。优点：不同光照下性能稳定、尺度不变，旋转不变，鲁棒性好；缺点：匹配成功数目少，速度较慢，有专利保护。目前在普通台式机上还无法实时计算SIFT特征，所以在对实时性要求较高的SLAM系统中用的很少。2006年提出了加速版，称为SURF。



SIFT特征点计算

FAST(Features from Accelerated Segment Test)

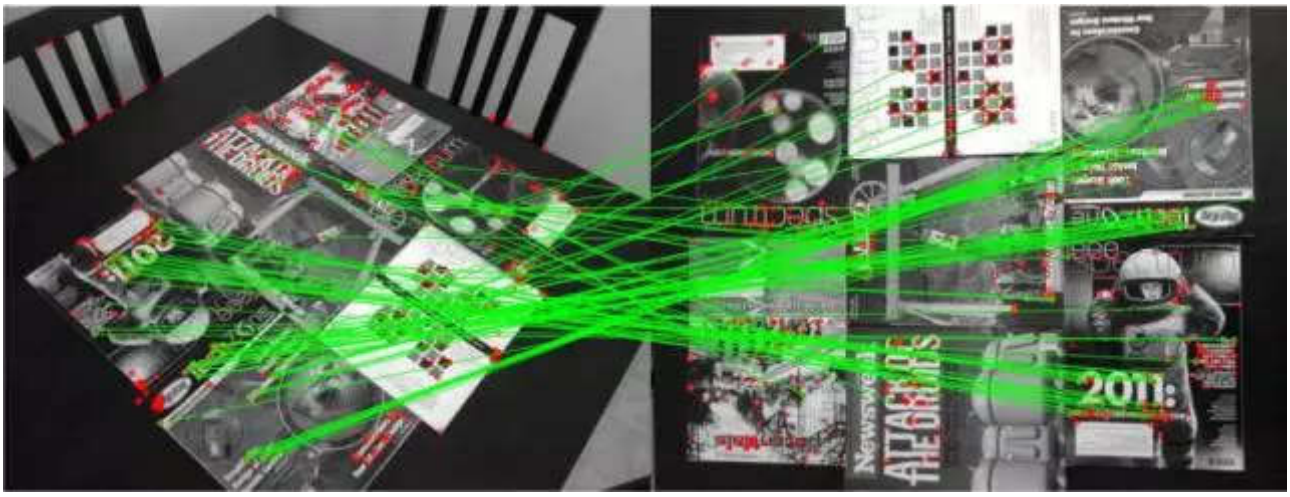
2006年提出。仅仅比较像素间亮度差，速度比较快。缺点：特征点数目庞大且不确定，不具备旋转不变性、尺度不变性。



FAST特征点

ORB (Oriented FAST and Rotated BRIEF)

2011年提出。是对FAST的改进，可以指定最终需要提取的特征点数量，具有旋转不变性和尺度不变性。采用速度极快的二进制描述子，速度仅为SURF的1/15。ORB在特征点质量和性能之间取得了较好的折中，在实时SLAM中非常受欢迎。



视角变化情况下的ORB特征点匹配结果。绿色连线表示有效的匹配对，红色圆圈表示未匹配的特征点。

2 特征匹配

根据上述的特征检测方法，我们可以分别得到相邻两张图中各自的特征点集合。特征匹配的目的就是把两张图中相同的特征点一一对应起来，如下图所示。

特征匹配示意图。红色为特征点，黄线表示匹配成功的特征点对

通常检测到的特征点都是成百上千个，那么这么多特征点如何进行匹配呢？

暴力匹配

最容易想到的办法就是对左图中的每一个特征点，计算它和右图所有特征点的相似程度（特征描述子的距离），然后按照相似程度进行排序，取最相似的那个作为匹配点，这称为暴力匹配。

暴力匹配虽然思想直观，但是有很明显的缺点：

- 1、计算复杂度是特征点数目的二次方。由于特征点一般数量巨大，所以该方法的计算量无法接受。
- 2、误匹配率高。由于特征点都是基于局部区域的特征，所以当场景中出现大量重复纹理（很常见）的时候很容易出现错误的匹配对。
-

由于上述原因，暴力匹配很少有人使用。

握手匹配

是对上述暴力匹配的改进。暴力匹配是左图的每个特征点和右图所有的特征点进行匹配，建立了多个匹配对（在此称为匹配对网）；然后握手匹配在这基础上又用右图每个特征点和左图的所有特征点进行匹配，也建立了新的匹配对网。这两个匹配对网的交集就是相互认为对方是最优选的匹配对应。我们称之为握手匹配。

握手匹配虽然减少了错误匹配的数量，但是计算复杂度至少是暴力匹配的2倍，仍然不实用。

快速近似最近邻匹配

快速近似最近邻匹配（FLANN）是一种适合于特征点数量极多情况下的方法，该方法比较成熟，且已经集成到OPENCV（开源计算机视觉库）中了，调用很方便。

匹配后的特征对可能包含一些错误匹配对，我们可以人为的设定一些筛选条件，剔除异常的匹配对。比如剔除距离（欧氏距离或者汉明距离）超过中值距离3倍以外的匹配对。

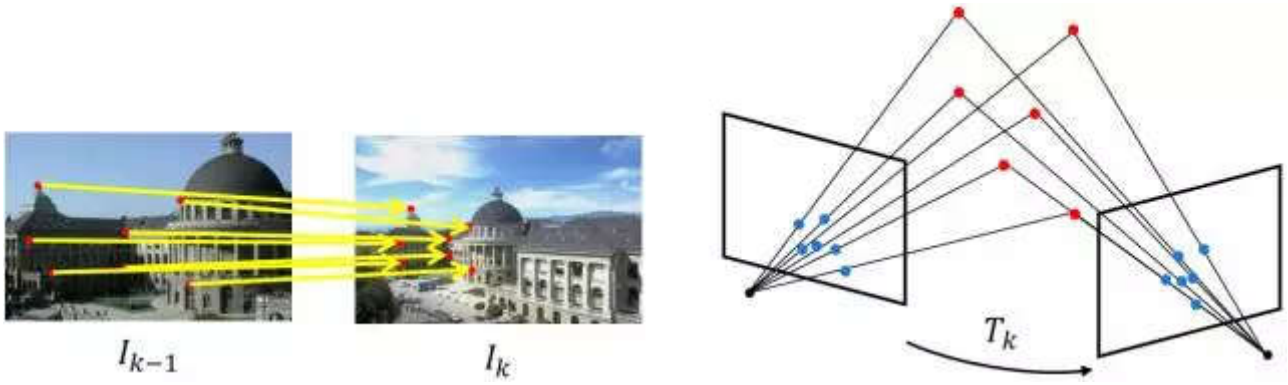
虽然经过了筛选，但是我们也不能保证每一个匹配对都是完全正确的，这其中仍然会存在错误的匹配。在后面的运动估计中，还需要去除错误匹配的方法RANSAC（随机采样一致性）。

3 运动估计

确定了特征匹配对，下一步就是根据这些特征匹配对估计相机的运动。根据 特征匹配对的不同维度（2D或3D），运动估计分为三种方法：

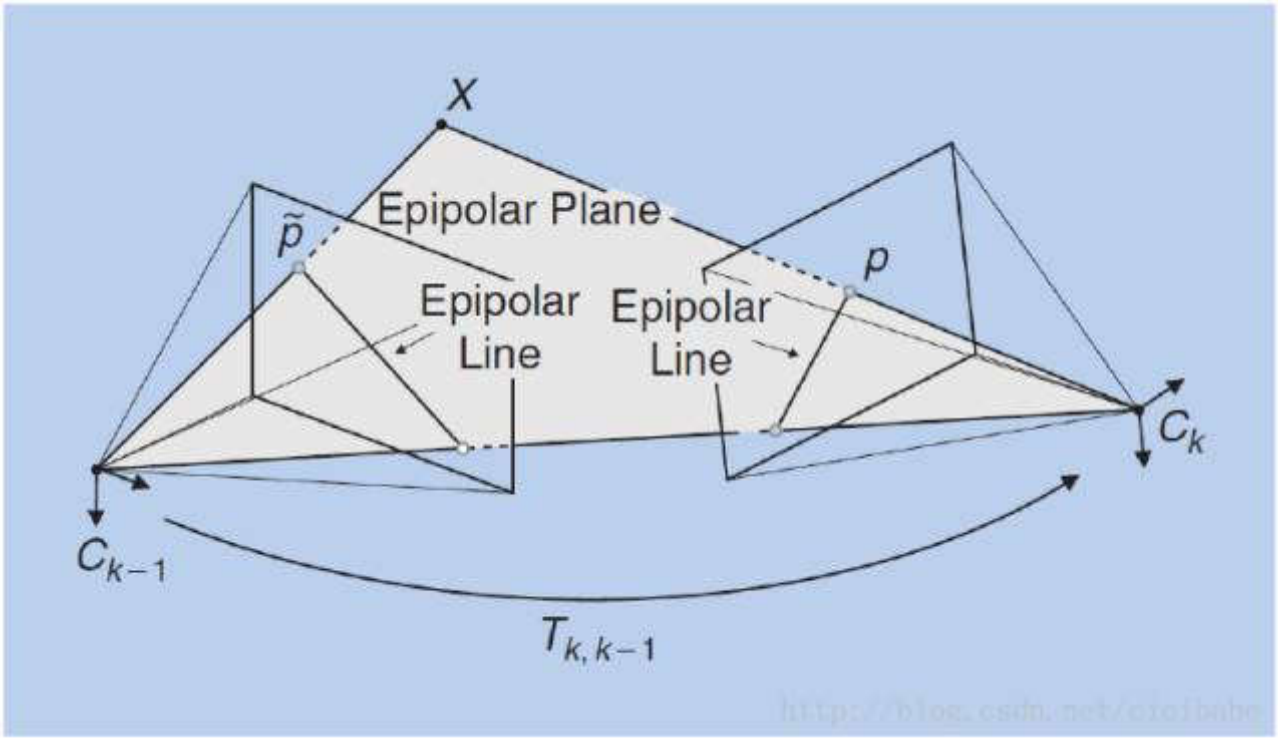
1、2D-2D

一般是单目RGB相机拍摄的相邻两帧图片，当我们得到了2D像素坐标的特征点匹配对，目标就是根据两组2D特征点对来估计相机的运动。该问题用**对极约束(极线约束)**来解决。



2D-2D运动估计示意图

如图所示，空间点X和两个相机中心点Ck-1、Ck形成了3D空间中的一个平面，称为极平面（epipolar plane）。极平面和两幅图像相交于两条直线，这两条直线称为对极线(epipolar line)。从图中看如果已知了正确的特征点匹配对p,p'，那么可以推断出三维点X的空间位置，以及相机的运动。



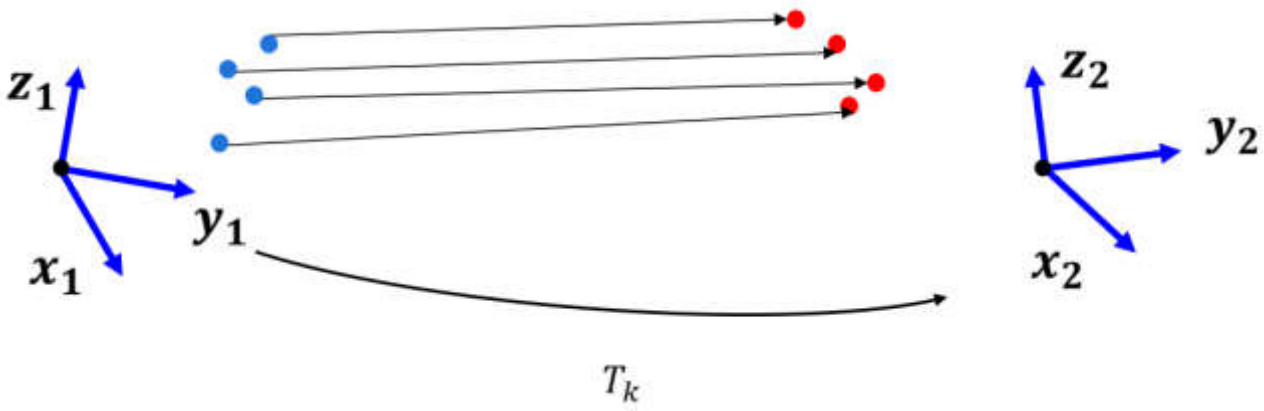
对极约束

具体求解过程：根据匹配好的2D点求出本征矩阵E或者基础矩阵F（一般使用八点法来求解）；然后根据E或者F求出旋转矩阵R和平移矩阵t，就得到了相机的位姿估计。

2D-2D的对极几何方法通常需要8个或8个以上的点对，且需要初始化（不能是纯旋转，必须有一定程度的平移）来解决尺度的不确定性。

2、3D-3D

双目相机或者RGB-D相机可以得到图像点的三维空间位置，这时匹配好的特征点对就是两组3D点，这种情况下通过迭代最近点（iterative closest point, ICP）来估计相机的运动。

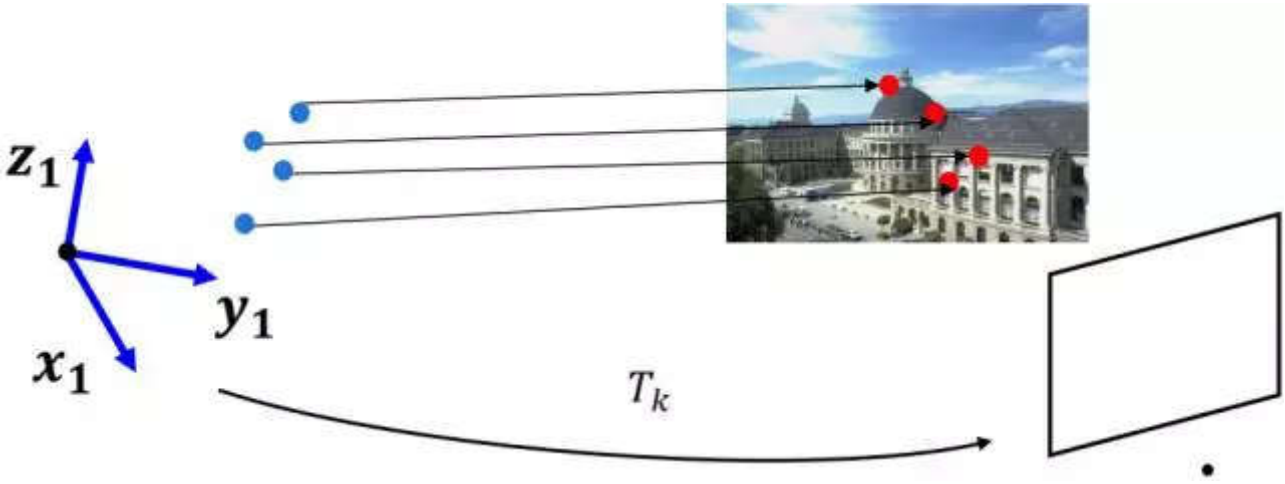


3D-3D示意图

3、3D-2D

当知道3D空间点及其在图像上的2D投影位置时，可以用PnP(perspective-n-point)来估计相机的位姿。至少需要3个点对（需要至少一个额外点验证结果）就可以估计相机的运动。

特征点的3D位置可以通过三角化或者RGB-D相机深度图确定，因此在双目或者RGB-D的VO中，可以直接使用PnP估计相机运动。而单目VO必须先初始化，才能使用PnP。3D-2D方法不需要使用对极约束，又可以在很少的匹配点中获得较好的运动估计，是最重要的一种姿态估计方法。



3D-2D示意图

上述三种运动估计的方法的适用条件如下所示：

对应关系	单目相机	双目相机	RGB-D相机
2D-2D	√	√	√
3D-3D	×	√	√
3D-2D	√	√	√

不同运动估计方法对比

4 离群点移除

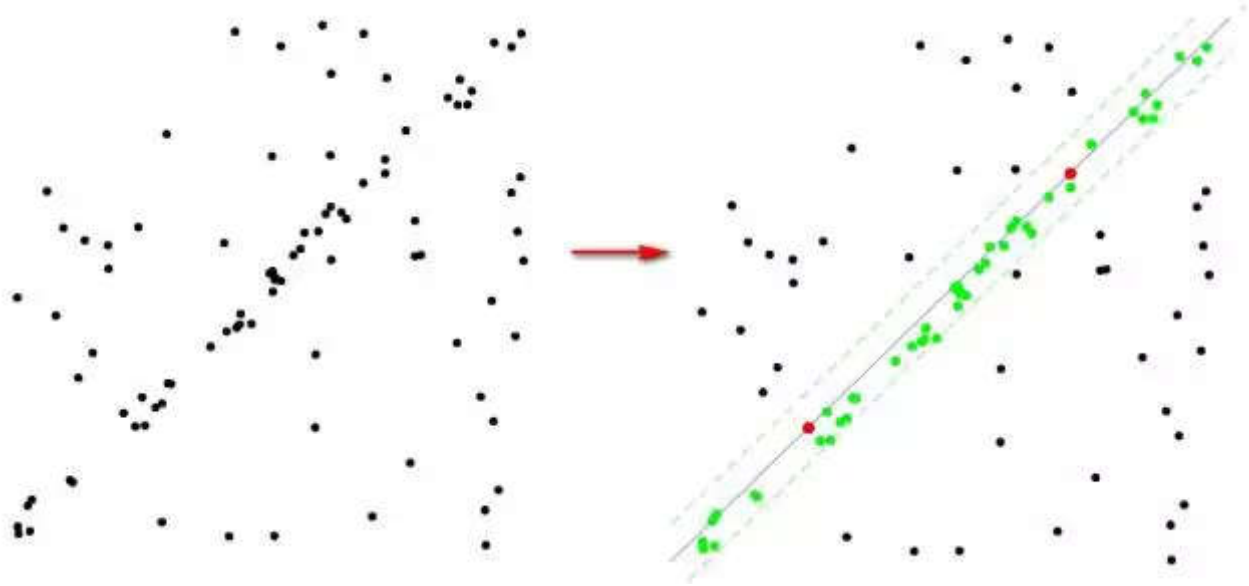
图像是现实三维世界的投影，在成像的过程中受到光照变化、视角变化、运动模糊等多种因素的影响，相邻图像可能呈现较大的差异。此时，匹配好的点对仍会夹杂不少错误的匹配，我们称之为离群点，这会造成错误的数据关联。如果需要相机运动准确地估计，那么移除离群点就非常重要。

最常用的离群点去除方法就是随机采样一致算法（RANdom SAmple Consensus，RANSAC），该方法对包含较多离群点的情况仍然适用。RANSAC的核心思想是多次随机提取数据点计算假设模型，再在其他数据点中验证这个假设。如果其他数据都一致验证通过这个假设模型，就认为该模型是真实的模型。

其流程如下：

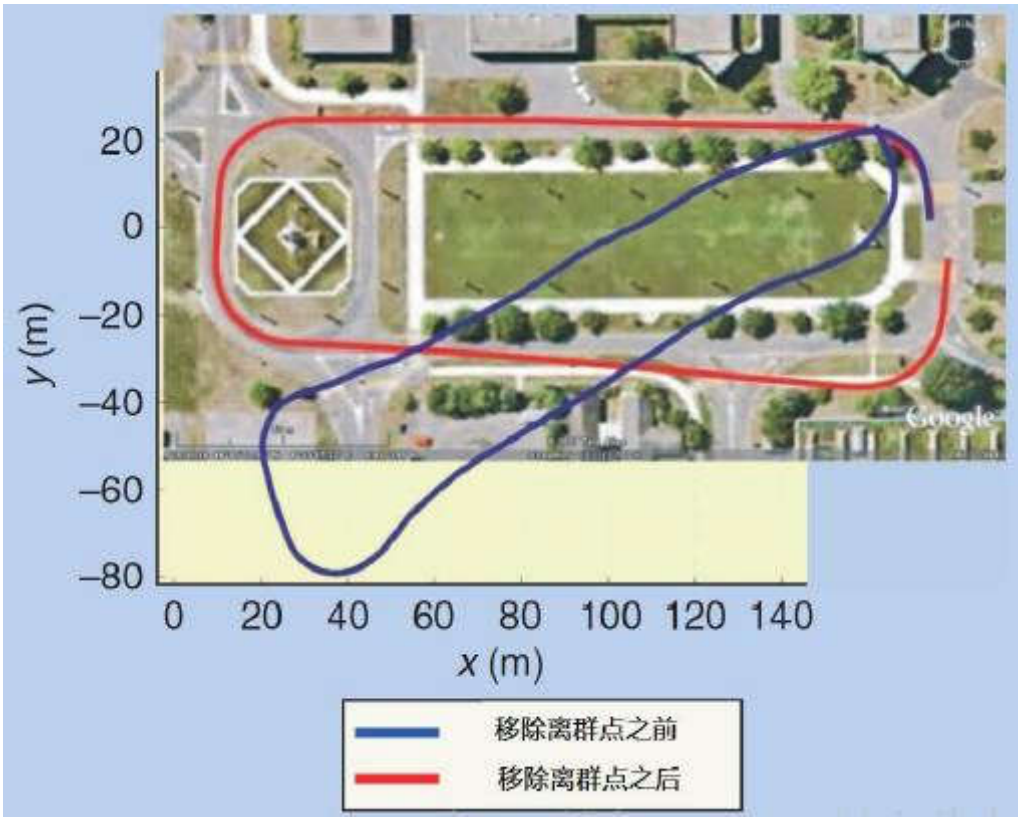
1) 初始化：A是一组N个特征的对应关系
2) 重复以下步骤：
    2.1) 从A中随机选取一组点s
    2.2) 给这些点适配一个模型
    2.3) 计算所有其他点到这个模型的距离
    2.4) 构建内点集合（比如，计算到模型距离<d的点的数量）
    2.5) 存储内点
    2.6) 迭代直到达到最大的迭代次数
3) 选择数量最多的一组内点作为方案
4) 用所有内点估计模型。

下图是RANSAC的一个示例，左边是所有的数据点，右边是RANSAC后的结果。



绿色和红色点是最后确定的内点，其他黑色点是离群点

下图显示了移除离群点前后的视觉里程计结果。

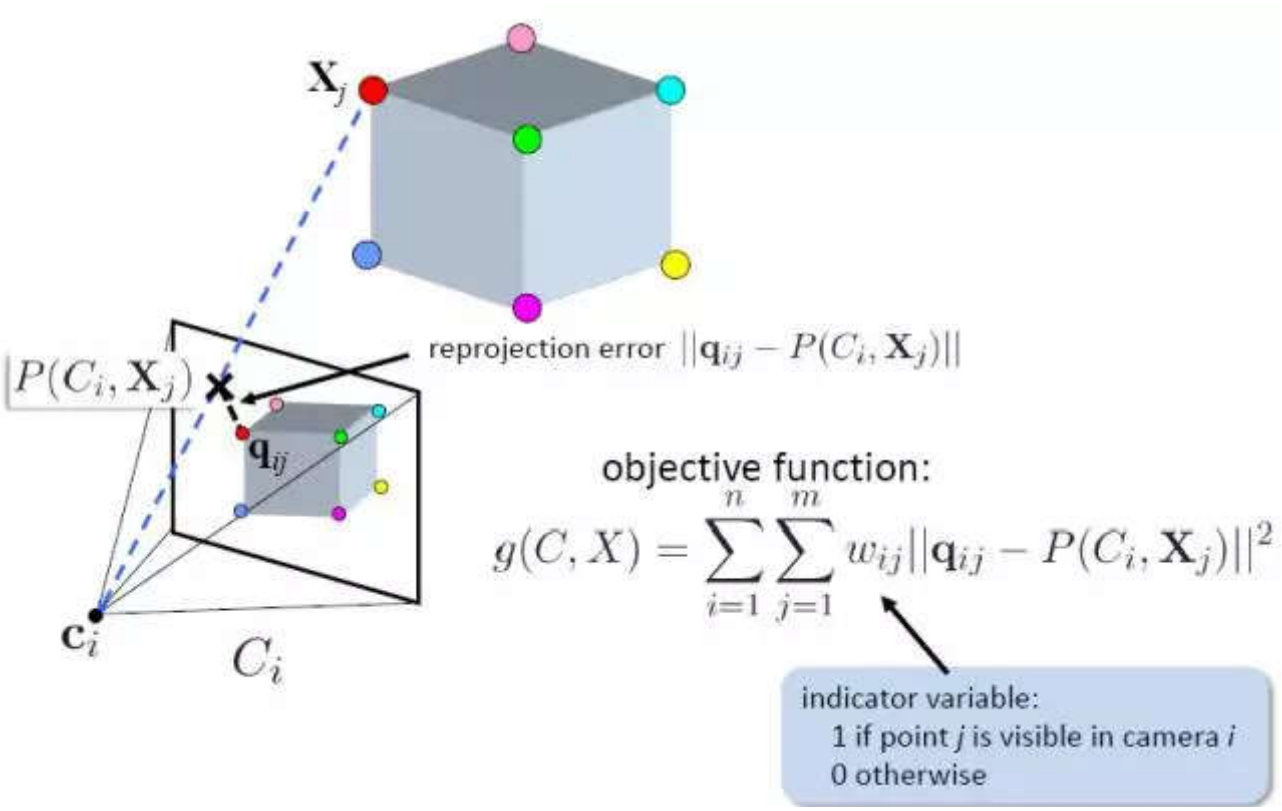


之前估计的视觉里程计轨迹与移除离群点之后的对比

5 Bundle Adjustment优化

Bundle Adjustment(BA)中文有多种翻译法：光束法平叉、束调整、捆集调整、捆绑调整等。BA的本质是一个优化模型，其目的是最小化重投影误差。什么是重投影误差呢?

如下图所示，空间物体通过相机在图像平面所成像就称之为投影。下图中红色的qij特征点就是空间三维立方体（不是图上方的那个立方体，图中未画出）在图像平面的投影。然后我们利用前面所述的运动估计方法可以估计qij对应的可能的三维空间点位置（Xj点，不一定是真实的位置）。然后我们利用估计的三维空间点Xj和计算的相机参数Ci（不一定是真值）重新进行投影，此时投影到图像中的位置为P(Ci,Xj)，由于估计位姿和相机参数时存在一定的误差，所以重投影点P(Ci,Xj)和第一次的投影qij有一定的偏差，就称为**重投影误差**。目标函数就是最小化这个重投影误差。



重投影误差示意图

而通常使用的是窗口BA，它将当前图像和之前的n-1幅图像帧作为一个窗口进行优化，这样不仅可以跟踪前一个相机位姿还可以跟踪更之前的相机位姿，保证当前和前n-1个相机位姿一致。因此，窗口BA可以减小漂移。窗口大小n的选择决定了计算复杂度，选择较小的窗口数量可以限制优化的参数的数量，使得实时BA成为可能。

以上就是特征点法视觉里程计的一个典型算法流程。下篇介绍另外一种VO方法：直接法。

相关阅读

SLAM初识
SLAM技术框架
惯性导航系统简介
视觉里程计：起源、优势、对比、应用