第七讲: 视觉里程计1

特征点法

slam分为视觉前端和视觉后端.前端也称为视觉里程计(VO). 它根据相邻图像的信息粗略估计相机的运动,给后端提供较好的初始值.

本讲学习如果提取匹配特征点,然后估计两帧之间的相机运动和场景结构.

特征点

图像本身是由亮度和色彩组成的矩阵.

特征点:

- 朴素角点: 简单,受环境,相机旋转等影响.
- 人工设计的特征点: 可重复性,可区别性,高效率性,本地性.

特征点是由关键点和描述子组成:

- 关键点: 图像中的位置
- 描述子: 附加信息,为了更好的区别其他点.

描述子通常是一个向量,按照某种认为设计的方式,描述了该关键点周围像素的信息.设计原则: 外观相似的特征应该有相似的描述子**.**

ORB(Oriented FAST and Rotated BRIEF) 是目前非常具有代表性的实时图像特征.

ORB是质量与性能较好的折中.

ORB特征

ORB由关键点和描述子组成.

关键点称为: Oriented FAST.是一种改进的FAST角点.

描述子称为: BRIEF(Binary Robust Independent Elementary Feature).

提取ORB特征点的步骤为:

- FAST角点提取: 找出图像中的角点. 相较于原版FAST,ORB中计算了特征点的主方向,为后续的BRIEF描述子增加了旋转不变性.
- BRIEF描述子: 对前一步中的特征点的周围进行描述.

FAST 关键点

FAST是一种角点,主要检测局部像素灰度变化明显的地方.特点是速度快.

$$ FAST特征点: \begin{cases} 优点: 仅比较像素间亮度的差异速度快 \ 缺点: \begin{cases} 特征点很多且不确定-决绝方法:指定要提取的角点数N,选取前N个具有最大响应值的点作为最终角点的集合 \ 不具有方向信息-决绝

方法: 添加旋转描述,其特征的旋转由灰度实心法实现 \ 存在尺度问题-决绝方法: 添加尺度描述,尺度不变性由构建图像金字塔,并在金字塔每层检验角点来实现. \end{cases} \end{cases} $$

质心:图像块灰度值作为质心.

操作步骤如下:

1. 在一个小的图像块B中,定义图像块的矩阵为:

$$ m_{pq} = \sum_{x,y \in B}x^py^qI(x,y),\qquad p,q={0,1} $$

2. 通过矩可以找到图像块的质心

$$ C = (\frac{m_{10}}{m_{00}},\frac{m_{01}}{m_{00}}) $$

3. 连接图形块的几何中心O与质心C, 得到一个方向向量$\overrightarrow{OC}$,于是特征点的方向可以定义为:

$$ \theta = \arctan(\frac{m_{01}}{m_{10}}) $$

通过以上信息,FAST便具有了尺度与旋转的描述.提升了健壮性. 在ORB中把这种改进后的FAST称为:Oriented FAST.

BRIEF描述子

提取关键点后对每个点计算其描述子.

BRIEF是一种二进制描述子. 例如,取关键点附近的p和q,如果p比q大,则取1,否则取0,如果去了128对这样的点,那么描述子可以用一个128位的二进制数来表示.

与原始的BRIEF相比,ORB描述子具有良好的旋转不变性.

特征匹配

特征匹配解决了SLAM中数据关联问题,即确定当前看到的路标与之前看到的路标之间的对于关系.

通过对图像与图像之间或者图像与地图之间的描述进行准确匹配,我们可以为后续姿态估计,优化等操作减轻大量负担.

如何两个图片中特征点集合的对应关系呢?

- 暴力匹配: 对每个特征点都测量描述子的距离,然后排序,取得最近的一个作为匹配点.

> 描述子的距离, 对于浮点型的描述子使用欧氏距离 对于二进制类型的描述子使用汉明距离

- 快速近似最邻近匹配: 更适合与匹配点的情况.它已经集成到了opencv当中.

实践: 特征提取和匹配

目前主流的几种特征提速方法在OpenCV中已经集成.

以下为openCV中图像特征提取,计算和匹配的过程.

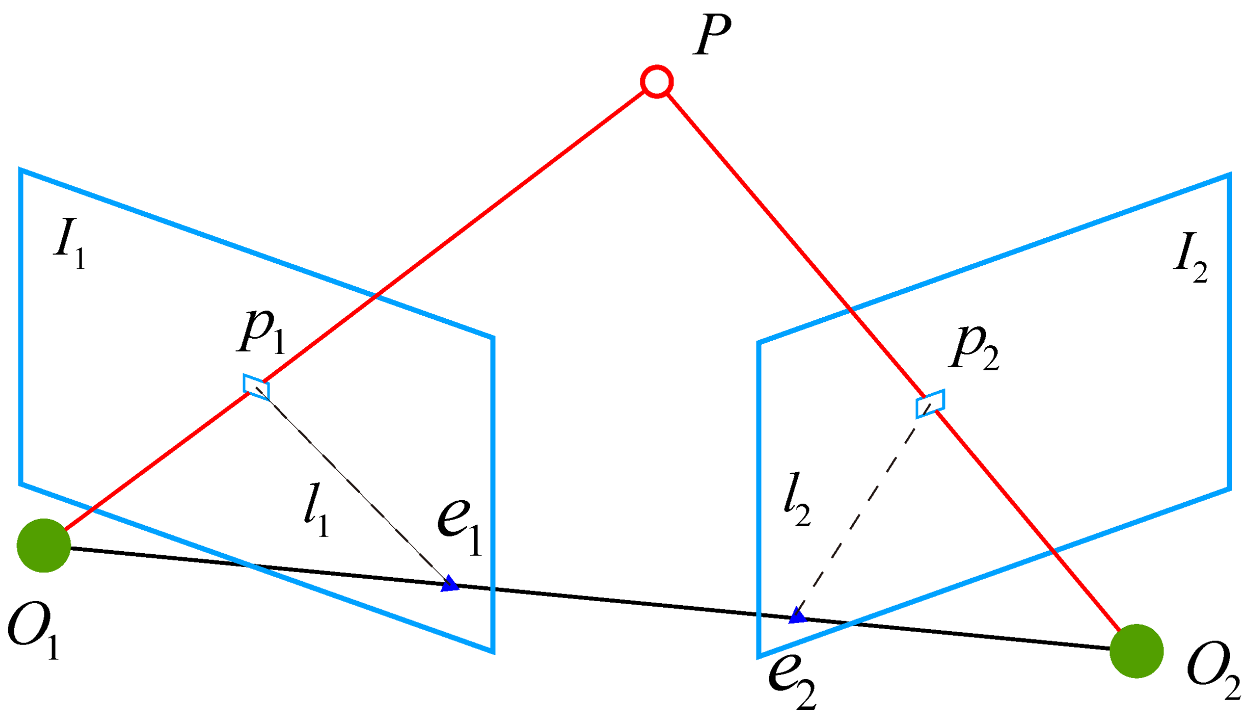
代码见:code\第七讲

我们希望根据匹配的点对来估计相机的运动.

1. 当相机为单目时,我们只知道2D的像素坐标,因而问题是根据两组2D点估计运动.该问题用对极几何来解决.
2. 当相机为双目,RGB等,即有距离信息. 那么就根据3D点估计运动. 常用ICP来解决.
3. 如果有3D点及在其他相机的投影位置,也能估计相机的运动. 该问题是通过PnP求解.

2D-2D:对极几何

对极约束



在第一帧的坐标系下,设P的空间位置为; $$ P = [X,Y,Z]^T $$

根据第五讲针孔相机模型,我们知道两个像素点p_1,p_2的像素位置为: $$ s_1p_1 = KP , \quad s_2p_2 = K(RP+t) $$

K为相机的内参矩阵. R,t为相机的运动,也可以写成李代数的形式.

如果使用齐次坐标,那么上式也可以写成如下形式; $$ p_1 = KP , \quad p_2 = K(RP+t) $$

取: $$ x_1 = K^{-1}p_1, \quad x_2 = K^{-1}p_2 $$

x_1,x_2是两个像素点 归一化平面的坐标.带入上式得: $$ x_2 = Rx_1 + t $$

两边同时左乘$t^\wedge$(^表示将向量变成矩阵),相当于两侧同时与t做外积: $$ t^\wedge x_2 = t^\wedge Rx_1 $$

然后同时左乘x_2^T $$ x_2^Tt^\wedge x_2 = x_2^T t^\wedge Rx_1 $$

因为$t^\wedge x_2 = 0$,所以上式化简为: $$ x_2^Tt^\wedge R x_1 = 0 $$

重新带入p_1,p_2得: $$ p_2^TK^{-T} t^\wedge RK^{-1} p_1 = 0 $$

以上两个式子都成为 对极约束.

它的几何意义是O_1,P,O_2三者共面. 对极约束中同时包含了 平移和旋转.

把中间部分记做两个矩阵:

- F : 基础矩阵
- E : 本质矩阵

于是进一步简化对极约束; $$ E = r^\wedge R , \quad F = K^{-T}EK^{-1}, \quad x_2^TEx_1 = p_2^TFp_1 = 0 $$

$对极约束简洁的给出了两个匹配点的空间位置关系.$ 于是相机的位姿估计可以分为以下步骤:

1. 根据匹配点的像素位置求出E 或者 F
2. 根据E或者F 求出 R,t

> 由于E和F只相差了相机内参,所以往往使用更简洁的EK

本质矩阵

本质矩阵$E=t^\wedge R$. 它是一个3×3矩阵. 从E的构造方式看,有以下几点需要注意:

1. 本质矩阵由对极约束定义.由于对极约束是等式为0的,所以对E乘以任意非零常数, 对极约束仍然满足. 我们把这件事称为E在不同尺度下是等价的.
2. 根据$E=t^\wedge R$,课证明本质矩阵E的奇异值必定是$[\sigma,\sigma,0]^T$的形式. 这称为 $本质矩阵的内在性质.$
3. 平移加旋转共有6个自由度,所以 $t^\wedge R$共有6个自由度.但是由于 尺度等价性, 实际上只有5个自由度.

使用8点法求解E.

```
graph LR
        a[E与t和R相关]
        b[t,R各有三个自由度]
        c[E有6个自由度]
        d[E为本质矩阵]
        e[R度不变性,去掉一个自由度]
        f[E剩下5个自由度]
        a-->b
        b-->c
        c-->d
        d-->e
        e-->f

        g[E当做普通矩阵,3*3,9个自由度]
        h[去掉R度不变性]
        i[剩下8个自由度]
        c-->g
        g-->h
        h-->i
```