

# 视觉SLAM：从理论到实践

## 第五次课 特征点法视觉里程计



主讲人 高翔

清华大学 自动控制与工程 博士  
慕尼黑工业大学计算机视觉组 博士后  
Email: [gao.xiang.thu@gmail.com](mailto:gao.xiang.thu@gmail.com)



# 第五讲 特征点法VO

1. 特征点提取与匹配
2. 2D-2D 对极几何
3. 3D-2D PnP
4. 3D-3D ICP
5. 三角化与深度估计
6. 实践穿插于各小节中间

# 1. 特征点提取与匹配

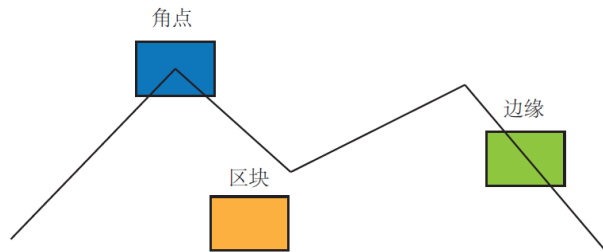
---

# 1. 特征点提取与匹配

- 经典SLAM模型中以位姿——路标（Landmark）来描述SLAM过程
  - 路标是三维空间中固定不变的点，能够在特定位姿下观测到
  - 数量充足，以实现良好的定位
  - 较好的区分性，以实现数据关联
- 在视觉SLAM中，可利用图像特征点作为SLAM中的路标

# 1. 特征点提取与匹配

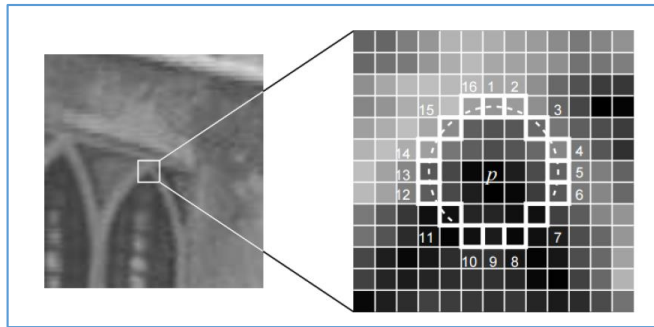
- 特征点：图像当中具有代表性的部分
  - 可重复性
  - 可区别性
  - 高效
  - 本地
- 特征点的信息
  - 位置、大小、方向、评分等——关键点
  - 特征点周围的图像信息——描述子（Descriptor）
- 例子：SIFT/SURF/ORB 见OpenCV features2d模块



特征描述应该在光照、视角发生少量变化时仍能保持一致

# 1. 特征点提取与匹配

- 例子：ORB特征
  - 关键点：Oriented FAST
  - 描述：BRIEF
- FAST
  - 连续N个点的灰度有明显差异
- Oriented FAST
  - 在FAST基础上计算旋转



1. 在一个小的图像块  $B$  中，定义图像块的矩为：

$$m_{pq} = \sum_{x,y \in B} x^p y^q I(x,y), \quad p, q = \{0, 1\}.$$

2. 通过矩可以找到图像块的质心：

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

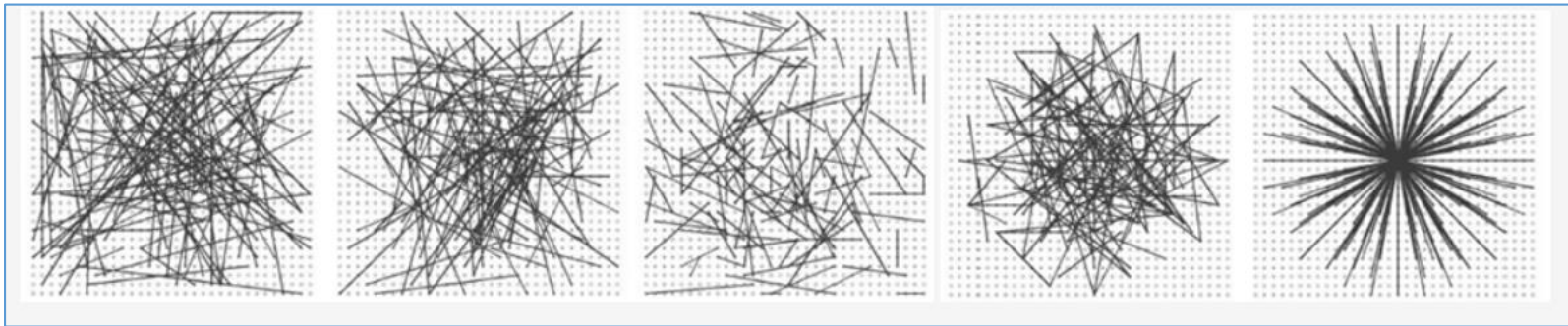
3. 连接图像块的几何中心  $O$  与质心  $C$ ，得到一个方向向量  $\overrightarrow{OC}$ ，于是特征点的方向可以定义为：

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

# 1. 特征点提取与匹配

- BRIEF

- BRIEF-128: 在特征点附近的128次像素比较



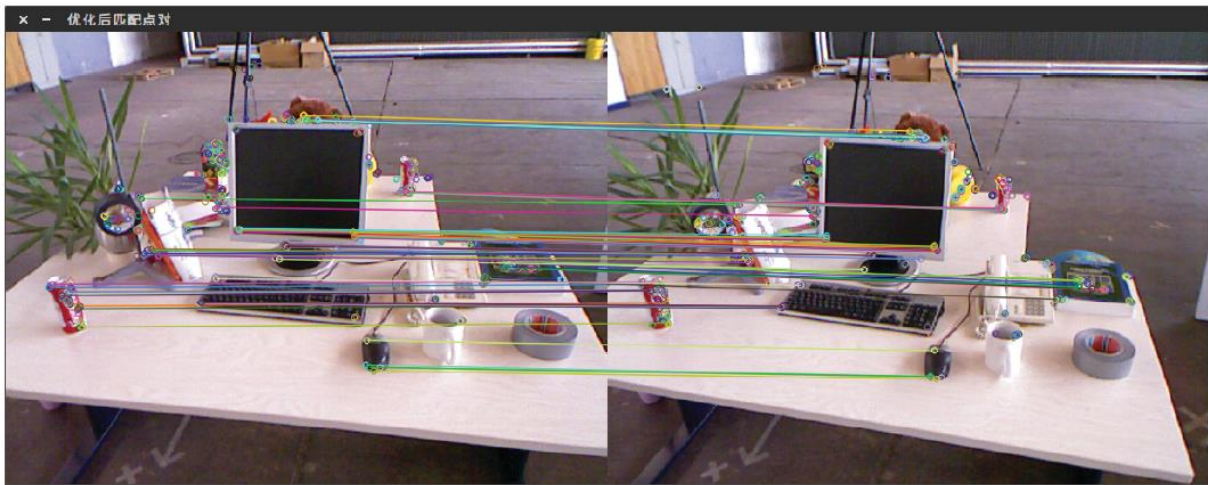
- ORB: 旋转之后的BRIEF描述
  - BRIEF是一种二进制描述，需要用汉明距离度量

BRIEF的比较pattern

# 1. 特征点提取与匹配

- 特征匹配

- 通过描述子的差异判断哪些特征为同一个点
- 暴力匹配：比较图1中每个特征和图2特征的距离
- 加速：快速最近邻（FLANN）





## 2. 2D-2D 对极几何

---

## 2. 2D-2D 对极几何

- 特征匹配之后，得到了特征点之间的对应关系
  - 如果只有两个单目图像，得到2D-2D间的关系 ——对极几何
  - 如果匹配的是帧和地图，得到3D-2D间的关系 ——PnP
  - 如果匹配的是RGB-D图，得到3D-3D间的关系 ——ICP

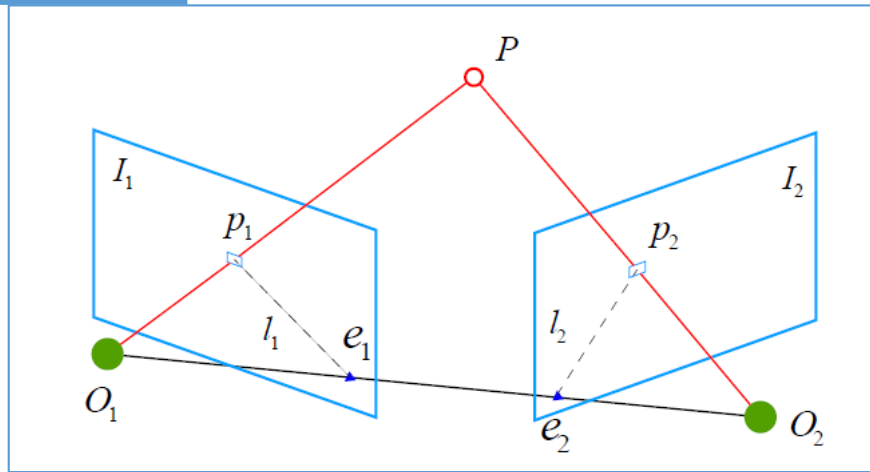
## 2. 2D-2D 对极几何

- 几何关系：

- $P$  在两个图像的投影为  $p_1, p_2$
- 两个相机之间的变换为  $T_{12}$
- $O_1P$  在第二个图像上投影为  $e_2p_2$ 
  - 记  $l_2$ ，称为极线，反之亦然
- $e_1, e_2$  称为极点

- 实践当中：

- $p_1, p_2$  通过特征匹配得到， $P$  未知， $e_1, e_2$  未知
- $T_{12}$  待求



## 2. 2D-2D 对极几何

- 世界坐标:  $P = [X, Y, Z]^T$ .
- 以第一个图作为参考系, 投影方程:

$$s_1 p_1 = K P, \quad s_2 p_2 = K (R P + t).$$

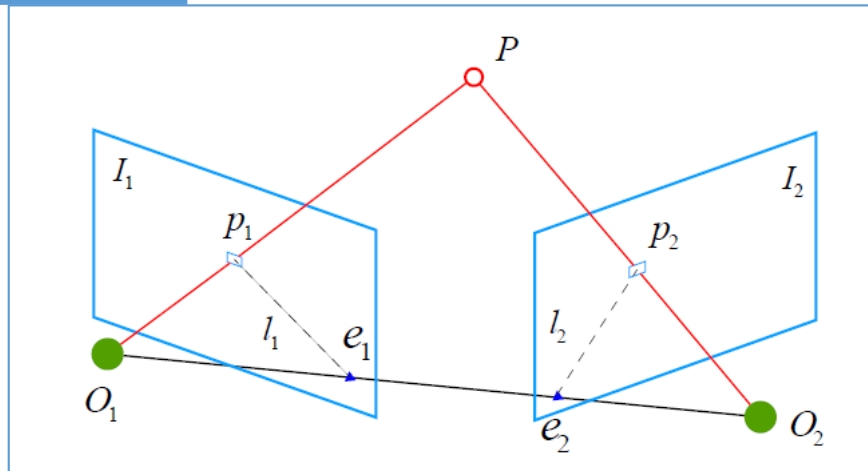
- 使用归一化坐标 (去掉内参):

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

- 齐次关系:  $x_2 = R x_1 + t.$

- 两侧左乘:  $t^\wedge x_2 = t^\wedge R x_1.$

- 再一步左乘:  $x_2^T t^\wedge x_2 = x_2^T t^\wedge R x_1.$



对极约束:

$$x_2^T t^\wedge R x_1 = 0.$$

带内参的形式:

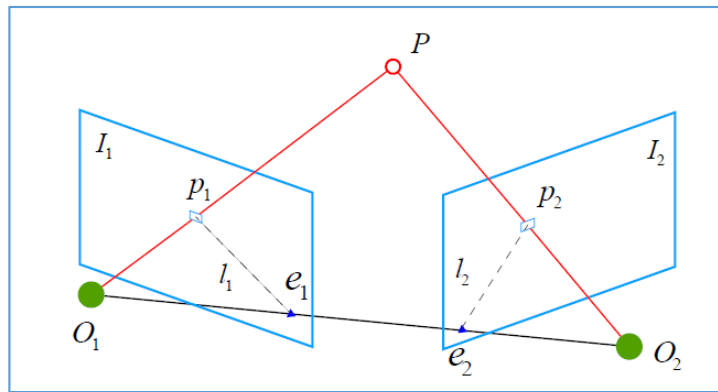
$$p_2^T K^{-T} t^\wedge R K^{-1} p_1 = 0.$$

## 2. 2D-2D 对极几何

- 对极约束刻画了  $O_1, O_2, P$  共面的关系
- 定义:

$$E = t^\wedge R, \quad F = K^{-T} E K^{-1}, \quad x_2^T E x_1 = p_2^T F p_1 = 0.$$

- Essential矩阵
- Fundamental矩阵
- 在内参已知的情况下, 可以使用E
- 两步计算位姿:
  - 由匹配点计算E
  - 由E恢复R,t



$$x_2^T t^\wedge R x_1 = 0.$$

$$p_2^T K^{-T} t^\wedge R K^{-1} p_1 = 0.$$

- 对极约束的性质:
  - 乘任意非零常数依然满足
  - E共五个自由度
    - 当成普通矩阵的话, 有八个自由度
    - 可用八点法求解

## 2. 2D-2D 对极几何

- 八点法求E
  - 将E看成通常3x3的矩阵，去掉因子后剩八个自由度
  - 一对匹配点带来的约束：

$$(u_1, v_1, 1) \begin{pmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{pmatrix} \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = 0.$$

- 向量形式：  $e = [e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9]^T$ ,

$$[u_1 u_2, u_1 v_2, u_1, v_1 u_2, v_1 v_2, u_2, v_2, 1] \cdot e = 0.$$

八对点构成方程组

$$\begin{pmatrix} u_1^1 u_2^1 & u_1^1 v_2^1 & u_1^1 & v_1^1 u_2^1 & v_1^1 v_2^1 & v_1^1 & u_2^1 & v_2^1 & 1 \\ u_1^2 u_2^2 & u_1^2 v_2^2 & u_1^2 & v_1^2 u_2^2 & v_1^2 v_2^2 & v_1^2 & u_2^2 & v_2^2 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_1^8 u_2^8 & u_1^8 v_2^8 & u_1^8 & v_1^8 u_2^8 & v_1^8 v_2^8 & v_1^8 & u_2^8 & v_2^8 & 1 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \\ e_9 \end{pmatrix} = 0.$$

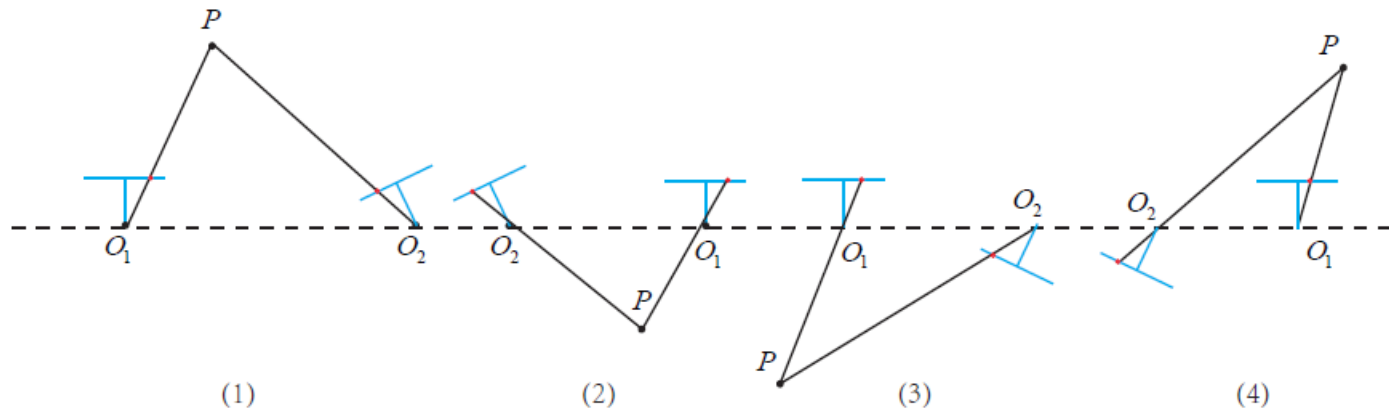
## 2. 2D-2D 对极几何

- 从E计算R, t: 奇异值分解

$$E = U\Sigma V^T,$$

$$\begin{aligned} t_1^\wedge &= UR_Z(\frac{\pi}{2})\Sigma U^T, & R_1 &= UR_Z^T(\frac{\pi}{2})V^T \\ t_2^\wedge &= UR_Z(-\frac{\pi}{2})\Sigma U^T, & R_2 &= UR_Z^T(-\frac{\pi}{2})V^T. \end{aligned}$$

四个可能的解，但只有一个深度为正



## 2. 2D-2D 对极几何

- SVD过程中：
  - 取  $E = U \text{diag}(\frac{\sigma_1 + \sigma_2}{2}, \frac{\sigma_1 + \sigma_2}{2}, 0) V^T$ . 因为E的内在性质要求它的奇异值为  $\sigma, \sigma, 0$
- 最少可使用五个点计算R,t, 称为五点法
  - 但需要利用E的非线性性质, 原理较复杂



## 2. 2D-2D 对极几何

---

- 八点法的讨论
  - 用于单目SLAM的初始化
  - 尺度不确定性：归一化  $t$  或特征点的平均深度
  - 纯旋转问题： $t=0$  时无法求解
  - 多于八对点时：最小二乘
  - 有外点时：RANSAC

## 2. 2D-2D 对极几何

- 从单应矩阵恢复R,t:

- 八点法在特征点共面时会退化

- 设特征点位于某平面上:  $n^T P + d = 0$ . 或  $-\frac{n^T P}{d} = 1$ .

- 两个图像特征点的坐标关系:

$$\begin{aligned} p_2 &= K(RP + t) \\ &= K\left(RP + t \cdot \left(-\frac{n^T P}{d}\right)\right) \\ &= K\left(R - \frac{tn^T}{d}\right)P \\ &= K\left(R - \frac{tn^T}{d}\right)K^{-1}p_1. \end{aligned}$$

中间记为H

$$p_2 = Hp_1.$$

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}.$$

## 2. 2D-2D 对极几何

- 该式是在非零因子下成立的

- 去掉第三行:

$$u_2 = \frac{h_1 u_1 + h_2 v_1 + h_3}{h_7 u_1 + h_8 v_1 + h_9}$$

$$v_2 = \frac{h_4 u_1 + h_5 v_1 + h_6}{h_7 u_1 + h_8 v_1 + h_9}$$

- 一对点提供两个约束
- 写成关于H的线性方程:
- 类似八点法
  - 先计算H
  - 再用H恢复R,t,n,d,K (8选1)

$$\begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} u_1^1 & v_1^1 & 1 & 0 & 0 & 0 & -u_1^1 u_2^1 & -v_1^1 u_2^1 \\ 0 & 0 & 0 & u_1^1 & v_1^1 & 1 & -u_1^1 v_2^1 & -v_1^1 v_2^1 \\ u_1^2 & v_1^2 & 1 & 0 & 0 & 0 & -u_1^2 u_2^2 & -v_1^2 u_2^2 \\ 0 & 0 & 0 & u_1^2 & v_1^2 & 1 & -u_1^2 v_2^2 & -v_1^2 v_2^2 \\ u_1^3 & v_1^3 & 1 & 0 & 0 & 0 & -u_1^3 u_2^3 & -v_1^3 u_2^3 \\ 0 & 0 & 0 & u_1^3 & v_1^3 & 1 & -u_1^3 v_2^3 & -v_1^3 v_2^3 \\ u_1^4 & v_1^4 & 1 & 0 & 0 & 0 & -u_1^4 u_2^4 & -v_1^4 u_2^4 \\ 0 & 0 & 0 & u_1^4 & v_1^4 & 1 & -u_1^4 v_2^4 & -v_1^4 v_2^4 \end{pmatrix} \begin{pmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \end{pmatrix} = \begin{pmatrix} u_2^1 \\ v_2^1 \\ u_2^2 \\ v_2^2 \\ u_2^3 \\ v_2^3 \\ u_2^4 \\ v_2^4 \end{pmatrix}$$

## 2. 2D-2D 对极几何

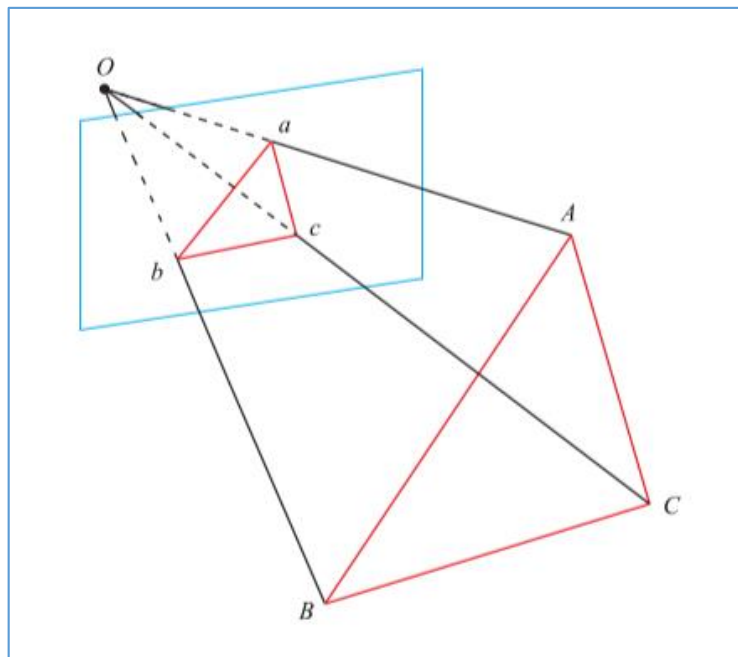
- 小结
- 2D-2D情况下，只知道图像坐标之间的对应关系
  - 当特征点在平面上时（例如俯视或仰视），使用H恢复 $R, t$
  - 否则，使用E或F恢复 $R, t$
  - $t$  没有尺度
- 求得 $R, t$ 后：
  - 利用三角化计算特征点的3D位置（即深度）
- 实际中用于单目SLAM的初始化部分

### 3. 3D-2D PnP

---

# 3. 3D-2D PnP

- 已知3D点的空间位置和相机上的投影点，求相机的旋转和平移（外参）
- 代数的解法/优化的解法
- 代数的
  - DLT
  - P3P
  - EPnP/UPnP/...
- 优化的：Bundle Adjustment



# 3. 3D-2D PnP

- DLT

- 设空间点  $P = (X, Y, Z, 1)^T$
- 投影点为:  $x = (u, v, 1)$  归一化坐标
- 投影关系:  $sx = [R|t]p$
- 展开:

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$

- 将它看成一个关于t的线性方程, 求解t

- 注意最下一行为

$$s = [t_9, t_{10}, t_{11}, t_{12}][X, Y, Z, 1]^T$$

- 用它消掉前两行中的s, 则一个特征点提供两个方程:

$$t_1 = (t_1, t_2, t_3, t_4)^T, t_2 = (t_5, t_6, t_7, t_8)^T, t_3 = (t_9, t_{10}, t_{11}, t_{12})^T,$$

$$t_1^T P - t_3^T P u_1 = 0, \quad t_2^T P - t_3^T P v_1 = 0.$$

- 为求解12个未知数, 需要12/2=6对点。  
(超定时求最小二乘解)

## 3. 3D-2D PnP

- DLT将R,t看成独立的未知量，所以在求出结果后，需要将t组成的矩阵投影回SO(3)（通常用QR分解实现）
- 此外，也可代入内参矩阵K，但SLAM中一般假设K已知，所以这里没有代入。
- 除DLT外：
  - P3P：利用三对点求相机外参
  - EPnP
  - .....

$$s \begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = \begin{pmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}.$$



# 3. 3D-2D PnP

- PnP的优化解法：Bundle Adjustment

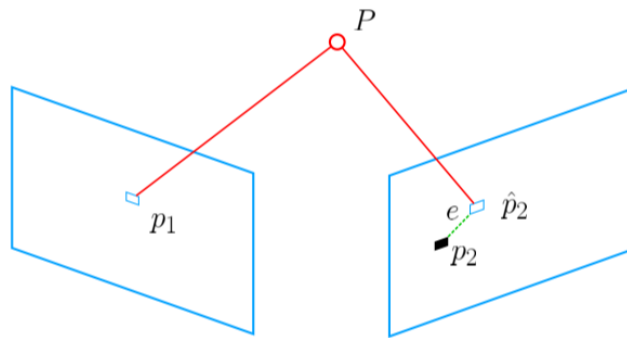
- 最小化重投影误差（Minimizing a reprojection error）

- 投影关系： $s_i u_i = K \exp(\xi^\wedge) P_i.$

- 定义重投影误差并取最小化：

$$\xi^* = \arg \min_{\xi} \frac{1}{2} \sum_{i=1}^n \left\| u_i - \frac{1}{s_i} K \exp(\xi^\wedge) P_i \right\|_2^2.$$

$P$ : 3d点,  $\xi$  相机位姿  $K$  内参  $u$  投影



# 3. 3D-2D PnP

- 线性化和雅可比

- 考虑单个投影点误差：

$$e_i = u_i - \frac{1}{s_i} K \exp(\xi^\wedge) P_i$$

- 线性化： $e_i(x + \Delta x) \approx e_i(x) + J(x)\Delta x$
- 雅可比的形式？

$$\frac{\partial e}{\partial \delta \xi} = \lim_{\delta \xi \rightarrow 0} \frac{e(\delta \xi \oplus \xi)}{\delta \xi} = \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial \delta \xi}.$$

第一项：

$$\begin{bmatrix} su \\ sv \\ s \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix}.$$

$$u = f_x \frac{X'}{Z'} + c_x, \quad v = f_y \frac{Y'}{Z'} + c_y.$$

$P'$ 为 $P$ 在相机坐标系下的坐标：

$$P' = (\exp(\xi^\wedge) P)_{1:3} = [X', Y', Z']^T.$$

对 $P'$ 进行投影：

$$su = K P'.$$

$$\frac{\partial e}{\partial P'} = - \begin{bmatrix} \frac{\partial u}{\partial X'} & \frac{\partial u}{\partial Y'} & \frac{\partial u}{\partial Z'} \\ \frac{\partial v}{\partial X'} & \frac{\partial v}{\partial Y'} & \frac{\partial v}{\partial Z'} \end{bmatrix} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix}.$$

### 3. 3D-2D PnP

$$\frac{\partial e}{\partial \delta \xi} = \lim_{\delta \xi \rightarrow 0} \frac{e(\delta \xi \oplus \xi)}{\delta \xi} = \frac{\partial e}{\partial P'} \frac{\partial P'}{\partial \delta \xi}.$$

- 第二项:  $\frac{\partial (TP)}{\partial \delta \xi} = (TP)^{\odot} = \begin{bmatrix} I & -P'^{\wedge} \\ 0^T & 0^T \end{bmatrix}.$

- 非齐次形式:  $\frac{\partial P'}{\partial \delta \xi} = [I, -P'^{\wedge}].$

- 两项相乘:

$$\frac{\partial e}{\partial \delta \xi} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} & -\frac{f_x X' Y'}{Z'^2} & f_x + \frac{f_x X^2}{Z'^2} & -\frac{f_x Y'}{Z'} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} & -f_y - \frac{f_y Y'^2}{Z'^2} & \frac{f_y X' Y'}{Z'^2} & \frac{f_y X'}{Z'} \end{bmatrix}.$$

### 3. 3D-2D PnP

- 也可以对3D点求导:

$$e_i = u_i - \frac{1}{s_i} K \exp(\xi^{\wedge}) P_i$$

$$\frac{\partial e}{\partial \mathbf{P}} = - \begin{bmatrix} \frac{f_x}{Z'} & 0 & -\frac{f_x X'}{Z'^2} \\ 0 & \frac{f_y}{Z'} & -\frac{f_y Y'}{Z'^2} \end{bmatrix} \mathbf{R}.$$

## 4. 3D-3D ICP

---

## 4. 3D-2D ICP

- 给定配对好的两组3D点，求其旋转和平移，可用迭代最近点（Iterative Closest Point, ICP求解）

- 设：
$$P = \{p_1, \dots, p_n\}, \quad P' = \{p'_1, \dots, p'_n\},$$

- 运动关系：
$$\forall i, p_i = Rp'_i + t.$$

- 同样定义误差项：
$$e_i = p_i - (Rp'_i + t).$$

- 以及最小二乘问题：

$$\min_{R, t} J = \frac{1}{2} \sum_{i=1}^n \|(p_i - (Rp'_i + t))\|_2^2.$$

## 4. 3D-2D ICP

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \| (p_i - (Rp_i' + t)) \|_2^2.$$

• 稍加推导:

1. 定义质心:

$$p = \frac{1}{n} \sum_{i=1}^n (p_i), \quad p' = \frac{1}{n} \sum_{i=1}^n (p_i').$$

2. 改写目标函数:

$$\begin{aligned} \frac{1}{2} \sum_{i=1}^n \| p_i - (Rp_i' + t) \|^2 &= \frac{1}{2} \sum_{i=1}^n \| p_i - Rp_i' - t - p + Rp' + p - Rp' \|^2 \\ &= \frac{1}{2} \sum_{i=1}^n \| (p_i - p - R(p_i' - p')) + (p - Rp' - t) \|^2 \\ &= \frac{1}{2} \sum_{i=1}^n (\| p_i - p - R(p_i' - p') \|^2 + \| p - Rp' - t \|^2 + \\ &\quad 2(p_i - p - R(p_i' - p'))^T (p - Rp' - t)). \end{aligned}$$

交叉项部分求和为零 =>

## 4. 3D-2D ICP

- 目标函数简化为：

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \|p_i - p - R(p_i' - p')\|^2 + \|p - Rp' - t\|^2.$$

只和R有关

和R,t都有关

- 最小化第一项，然后取t，使得后一项为零。
- 左侧项如何求？



# 4. 3D-2D ICP

$$\min_{R,t} J = \frac{1}{2} \sum_{i=1}^n \|p_i - p - R(p'_i - p')\|^2 + \|p - Rp' - t\|^2.$$

- 旋转的求取

- 定义去质心坐标:

$$q_i = p_i - p, \quad q'_i = p'_i - p'.$$

- 最小化:

$$R^* = \arg \min_R \frac{1}{2} \sum_{i=1}^n \|q_i - Rq'_i\|^2.$$

- 推导:

$$\frac{1}{2} \sum_{i=1}^n \|q_i - Rq'_i\|^2 = \frac{1}{2} \sum_{i=1}^n q_i^T q_i + q_i'^T R^T R q'_i - 2q_i^T R q'_i.$$

- 最右一项:

$$\sum_{i=1}^n -q_i^T R q'_i = \sum_{i=1}^n -\text{tr}(R q'_i q_i^T) = -\text{tr}\left(R \sum_{i=1}^n q'_i q_i^T\right).$$

- SVD解法:

$$W = \sum_{i=1}^n q_i q_i'^T. \quad W = U \Sigma V^T. \quad R = UV^T.$$

## 4. 3D-2D ICP

- ICP也可以从非线性优化角度求解，但：
  - 已知匹配时，ICP问题存在唯一解或无穷多解的情况。在唯一解的情况下，只要能找到极小值解，那么这个极小值就是全局最优值。
  - 所以正常情况下，SVD结果和优化一样，且优化很快收敛。
- 注：
  - 在激光情况下，匹配点未知，将指定最近点为匹配点。此时问题非凸，极小值不一定为最小值。
  - 利用非线性优化可以将ICP与PnP结合在一起求解。

## 5. 三角化与深度估计

---

# 5. 三角化与深度估计

- 已知运动时，求解特征点的3D位置

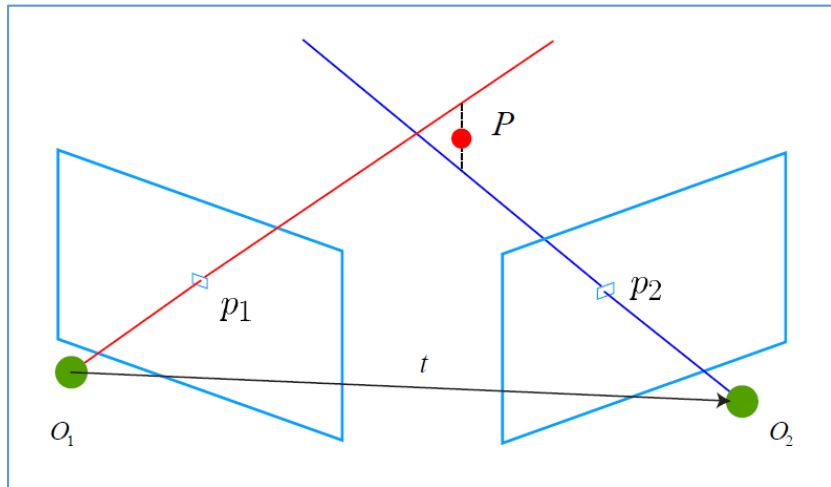
- 几何关系： $s_1 x_1 = s_2 R x_2 + t.$

- 求  $s_2$  时，两侧乘  $x_1^\wedge$

$$s_1 x_1^\wedge x_1 = 0 = s_2 x_1^\wedge R x_2 + x_1^\wedge t.$$

- 反之亦然
- 或者同时解  $s_1, s_2$  :

- 求  $[-R x_2, x_1] \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = t$  的最小二乘解  $x = (A^T A)^{-1} A^T b$



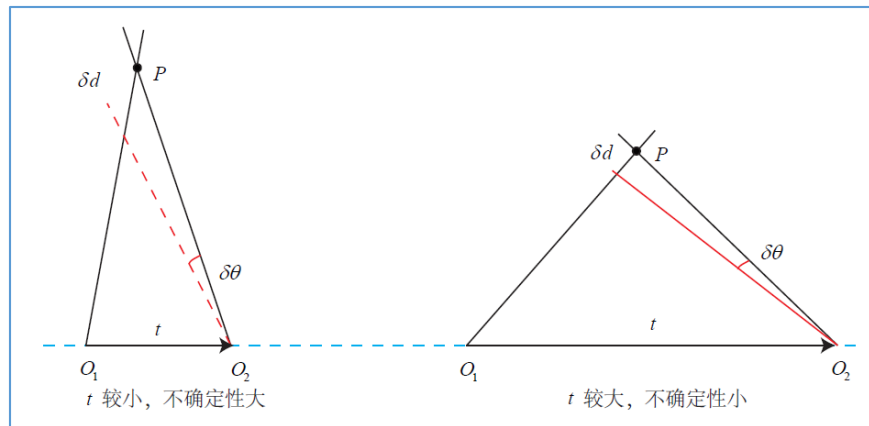
# 5. 三角化与深度估计

- 三角化中的问题：
  - 解得深度的质量与平移相关
  - 但是平移大时特征匹配可能不成功

- 方程 
$$\begin{bmatrix} -Rx_2 & x_1 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} = t$$

- 系数矩阵伪逆不可靠
- $A^T A$  行列式近零

- 例如：相机前进时，虽然有位移，但位于图像中心的点无法三角化（没有视差）



# 5. 三角化与深度估计

---

- 小结：
- 本章介绍了与特征点相关的视觉里程计部分算法，包括：
  - 特征点是如何提取并匹配的；
  - 如何通过2D-2D的特征点估计相机运动；
  - 三角化原理；
  - 3D-2D的PnP问题，线性解法与BA解法；
  - 3D-3DICP问题，线性解法与BA解法。