视觉里程计

7.1 特征点法

视觉里程计(VO)，**它根据相邻图像的信息，估计出粗略的相机运动，给后端提供较好的初始值**。VO的实现方法，按是否需要提取特征，分为特征点法的前端以及不提特征的直接法前端。基于特征点法的前端被认为是视觉里程计的主流方法。

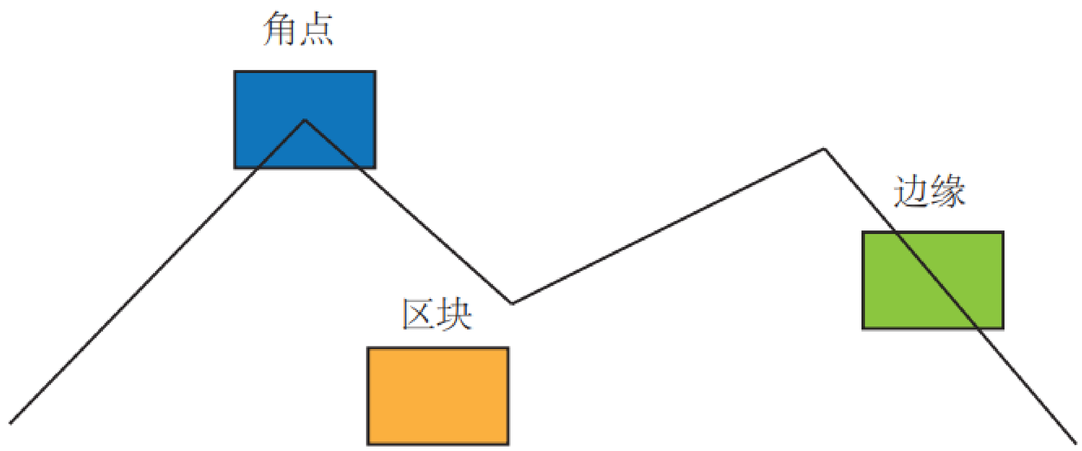
在此章节，我们从特征点法入手，学习如何提取、匹配图像特征点，然后估计两帧之间的相机运动和场景结构，从而实现一个基本的两帧视觉里程计。

7.1.1 特征点

首先，从图像中选取比较有**代表性的点**。这些点在相机视角发生少量变化后会保持不变，因此我们也会在其他图像中找到相同的点。然后，在这些点的基础上，**讨论相机位姿估计问题，以及点的定位问题**。

在视觉SLAM中，**以上有代表性的点被称为图像特征**，**特征是图像信息的另一种数字表达形式**。在视觉里程计中，我们希望：**特征点在相机运动之后保持稳定**。我们主要做的是**当场景和相机视角发生少量改变时，从图像中判断哪些地方是同一个点**。因此我们要进行特征点提取。

特征点提取：图像中，可以有角点，边缘和区块，这些可以当成图像中具有代表性的部分，如下图所示。



其中，角点辨识度最强。**我们需要的是设计更加稳定的局部图像特征！**(例如：SIFT，SURF，ORB)，这些人工设计的特征点具有非常好的特性：

1. 可重复性(Repeatability)：相同的“区域”可以在不同的图像中被找到
2. 可区别性(Distinctiveness)：不同的“区域”有不同的表达
3. 高效率(Efficiency)：同一图像中，特征点的数量应远小于像素的数量
4. 本地性(Locality)：特征仅与一小片图像其余相关

特征点由**关键点(Key-point)**和**描述子(Descriptor)**两部分组成。关键点是指该特征点在图像里的位置（以及或朝向、大小等信息）；描述子通常是一个向量，按照某种人为设计的方式，描述了该关键点周围像素的信息。

经典图像特征：SIFT(尺度不变特征变换，Scale-invariant Feature Transform)是经典的一种图像特征，它充分考虑了图像变换过程中出现的光照，尺度，旋转等变化。

P136

7.1.2 ORB特征

ORB特征是一种改进的FAST角点，它也是由**关键点和描述子两部分构成**。它的描述子称为BRIEF(Binary Robust Independent Elementary Features)。因此，提取ORB特征分为两个步骤：

① FAST角点提取：找出图像中的"角点"。

② BRIEF描述子：对前一步提取出特征点的周围图像区域进行描述。

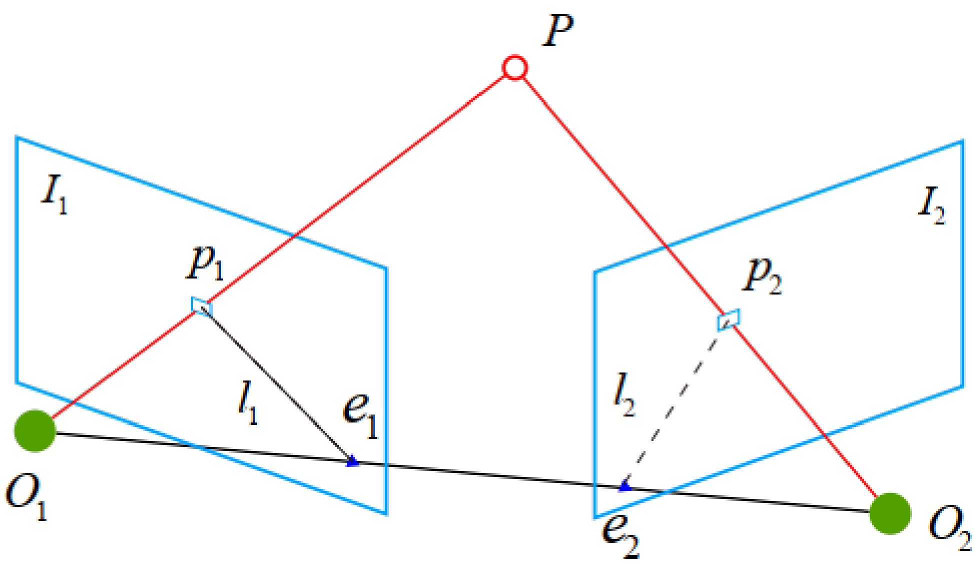
FAST是一种角点，主要检测局部像素灰度变化明显的地方，速度较快。**检测思想：如果一个像素与它邻域的像素差别较大，那它更可能是角点。**

7.1.3 特征匹配

7.2 2D-2D:对极几何

7.2.1 对极约束

假设我们从两张图中，得到了一对配对好的特征点，如下图所示：



如果我们有若干对这样的匹配点，就可以通过这些二维图像点的对应关系，恢复出在两帧之间摄像机的运动。

以上图为例，我们希望求取I1和I2之间的运动，设第一帧到第二帧的运动为R，t。两个相机中心分别为O1，O2。现在，考虑I1中有一个特征点p1，它在I2中对应着特征点p2。**若匹配正确，说明它们确实是同一个空间点在两个成像平面上的投影。**

接下来，我们介绍一些术语：连线O1p1和连线O2p2在三维空间中会相交于点P，O1O2P构成的平面称为**极平面(Epipolar plane)**。O1O2连线与相平面I1，I2的交点分别为e1，e2。e1，e2称为**极点(Epipoles)**，O1O2被称为**基线(Baseline)**。称极平面与两个相平面I1，I2之间的相交线为**极线(Epipolar line)**。

重点，2D-2D对极几何：

首先：我们设点P的空间位置为：

$$P = [X, Y, Z]^T$$

其次，由针孔相机模型，我们得到：

$$z_1 p_1 = KP$$
$$z_2 p_2 = K(RP + t)$$

其中，R和t是我们要求的，分别表示旋转矩阵和平移向量，K为相机内参。接下来，我们在归一化平面上表示像素点的坐标（把P的z化为1），有：

$$p_1 = KP$$
$$p_2 = K(RP + t)$$

现在取：

$$x_1 = K^{-1} p_1$$
$$x_2 = K^{-1} p_2$$

代入上式得到：

$$x_2 = Rx_1 + t$$

为了消去t，我们左乘t_hat

$$\hat{t}\, x_2 = \hat{t}\, Rx_1 + \hat{t}\, t = \hat{t}\, Rx_1 + 0$$

再左乘x2T，得到：

$$x_2^T \hat{t}\, x_2 = x_2^T \hat{t}\, Rx_1$$

注意到，t_hat与x2的乘积可以看作t与x2叉乘，因此这个向量与t和x2都垂直，因此等式左侧等于0，因此有

$$x_2^T \hat{t}\, Rx_1 = 0$$

重新带入p1，p2有：

$$p_2^T K^{-T} \hat{t}\, RK^{-1} p_1 = 0$$

这就是对极约束。我们定义两个矩阵：

$$E = \hat{t}\, R$$
$$F = K^{-T} E K^{-1}$$

F称为基础矩阵(Fundamental Matrix)，E称为本质矩阵(Essential Matrix)。

对极约束简洁地给出了两个匹配点的空间位置关系，因此相机位姿估计问题变为：

1. 根据配对点的像素位置，求出矩阵E或者矩阵F
2. 根据矩阵E或矩阵F，求出R，t

下面介绍如何求解

7.2.2 本质矩阵

研究本质矩阵的特性：

- 对本质矩阵乘以任意非零常数后，对极约束依然满足。
- 根据E=t^R可以证明，本质矩阵E的奇异值必定为[σ，σ，0]^T的形式。

- 由于平移和旋转各有三个自由度，因此E有六个自由度；但由于尺度等价性，故E实际上有五个自由度。

由于只用五个点求解困难，我们只考虑本质矩阵E的第一条特性，因此**八点法(Eight-point-algorithm)非常经典。**

八点法

考虑一对匹配点，（利用其归一化坐标），那么有：

$$\begin{bmatrix} u_1 & v_1 & 1 \end{bmatrix} \begin{bmatrix} e_1 & e_2 & e_3 \\ e_4 & e_5 & e_6 \\ e_7 & e_8 & e_9 \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = 0$$

我们利用矩阵乘法原理展开，可以得到：

$$u_1 u_2 e_1 + v_1 u_2 e_4 + u_2 e_7 + u_1 v_2 e_2 + v_1 v_2 e_5 + v_2 e_8 + u_1 e_3 + v_1 e_6 + e_9 = 0$$

我们再改写成矩阵乘法形式，并且把所有八个点都放到一个方程中，得到线性方程组：（ui^1表示第i个特征点）

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

这八个方程构成了一个线性方程组。它的系数矩阵由特征点位置构成，大小为8×9。e位于该矩阵的零空间中。若系数矩阵是满秩的(r=8)，则它的零空间维数为1，即基础解系一维，这样就可以解出E的各个元素。

接下来探讨如何根据矩阵E恢复出相机的运动R，t。 *(矩阵的奇异值分解(SVD)介绍在另一个文档里)*

设E的SVD分解为：

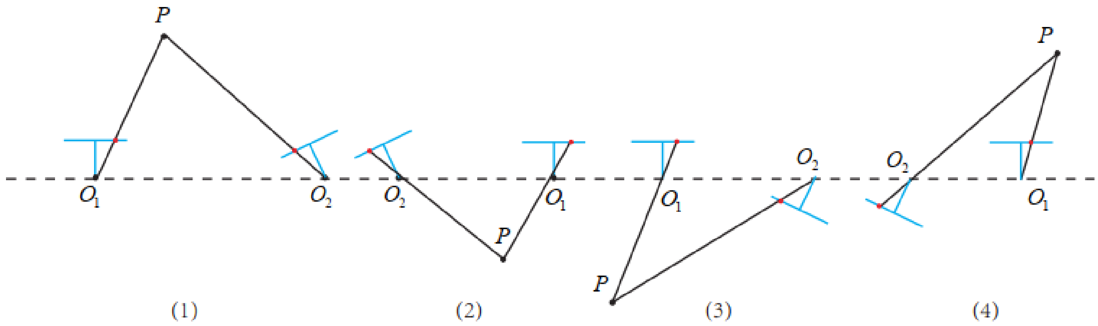
$$E = U \Sigma V^T$$

其中，U，V为正交矩阵，Σ为奇异值矩阵。根据E的内在性质，我们知道Σ=diag(σ，σ，0)。

在SVD分解中，对于任意一个E，存在两个可能的t，R与它对应：

$$\begin{aligned} \hat{t}_1 &= U R_Z(\pi/2) \Sigma U^T \\ \hat{t}_2 &= U R_Z(-\pi/2) \Sigma U^T \\ R_1 &= U R_Z^T(\pi/2) \Sigma V^T \\ R_2 &= U R_Z^T(-\pi/2) \Sigma V^T \end{aligned}$$

其中，Rz(pi/2)表示沿着Z轴旋转90°得到的旋转矩阵。同时，由于-E和E等价，所以对于任意一个t取负号，也会得到同样的结果。因此，从E分解到t，R时，一共存在四个可能的解，如下图所示。我们去深度为正的即可。



(1) (2) (3) (4)

当根据线性方程解出的E，可能不满足E的内在性质——它的奇异值不一定为σ，σ，0的形式。这是，我们假设Σ=diag(σ1，σ2，σ3)，依次增大，取：

$$E = U\begin{bmatrix} \frac{\sigma_1+\sigma_2}{2} & 0 & 0 \\ 0 & \frac{\sigma_1+\sigma_2}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

7.2.3 单应矩阵

除了基本矩阵F和本质矩阵E，还有一种单应矩阵H，它描述了两个平面之间的映射关系。**若场景中的特征点都落在同一平面上（比如墙，地面等等）**，则可以通过单应性来进行运动估计。

单应矩阵通常描述处于共同平面上的一些点，在两张图像之间的变换关系。考虑在图像I1和I2有一对匹配好的特征点p1和p2，这些特征点落在某平面上。设这个平面方程为：

$$n^T P + d = 0$$
$$d\text{为常数}$$

稍加整理：

$$-\frac{n^T P}{d} = 1$$

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

我们把中间部分记作H，这就是单应矩阵。展开我们可以得到：

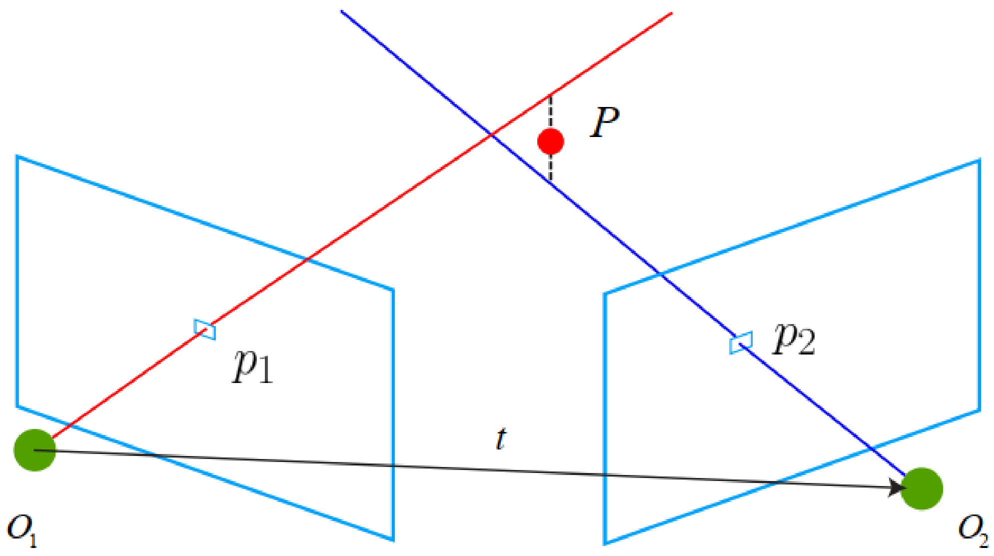
$$\begin{bmatrix} u_2 \\ v_2 \\ 1 \end{bmatrix} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix}$$

P152

7.5 三角测量

我们现在来用相机的运动估计特征点的空间位置。在单目SLAM中，仅通过单张图像无法获得像素深度信息，我们需要通过**三角测量(Triangulation)**的方法来估计地图点的深度。

三角测量指通过在两处观察同一个点的夹角，确定该点的距离。如下图所示：



设x1，x2为两个特征点的归一化坐标，那么有：

$$s_2x_2 = s_1Rx_1 + t$$

我们已知x1，x2，R，t，要求s1，s2。可以分开求解：

$$s_2\hat{x_2}x_2 = s_1\hat{x_2}Rx_1 + \hat{x_2}t = 0$$

这样可以通过求解上述方程解出s1。s2同理。

由于噪声的存在，估计得到的R，t不一定精确使上式为0，因此更常见的做法是求最小二乘解而不是零解（求取等式左侧的模长，求s1使得模长最小）。

7.7 3D-2D：PnP

PnP(Perspective-n-Point)是求解3D到2D点对运动的方法。它描述了当我们知道n个3D空间点以及它们的投影位置时，如何估计相机所在的位姿。

7.7.1 直接线性变换

考虑某个空间点P(X,Y,Z)已知，且其在图像中投影到特征点x1=(u1,v1,1)^T已知，深度信息s1未知，相机位姿(R,t)未知，我们根据单孔相机成像规律得到：

$$ZX = TP$$

$$s_1\begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = \begin{bmatrix} t_1 & t_2 & t_3 & t_4 \\ t_5 & t_6 & t_7 & t_8 \\ t_9 & t_{10} & t_{11} & t_{12} \end{bmatrix}\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

其中，T矩阵是增广矩阵：[R|t]，就是未知量。我们要求出这十二个未知量才可以。

推导：首先，我们把深度信息s消去(3个等式变为2个)f

$$su_1 = \overrightarrow{t_1}\overrightarrow{P}$$
$$sv_1 = \overrightarrow{t_2}\overrightarrow{P}$$
$$s = \overrightarrow{t_3}\overrightarrow{P}$$

③式子代入①、②即可得到：

$$\overrightarrow{t_1}\overrightarrow{P} - \overrightarrow{t_3}\overrightarrow{P}u_1 = 0$$
$$\overrightarrow{t_2}\overrightarrow{P} - \overrightarrow{t_3}\overrightarrow{P}v_1 = 0$$

我们对上两式左右取转置得到：

$$\overrightarrow{P^T}\overrightarrow{t_1^T} - \overrightarrow{P^T}\overrightarrow{t_3^T}u_1 = 0$$
$$\overrightarrow{P^T}\overrightarrow{t_2^T} - \overrightarrow{P^T}\overrightarrow{t_3^T}v_1 = 0$$

其中，ti向量为增广矩阵的行向量，转置就是列向量。把上两式写为矩阵形式，可以得到：

$$\begin{bmatrix} P^T & 0 & -u_1 P_1^T \\ 0 & P^T & -v_1 P_1^T \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ \dots \\ t_{12} \end{bmatrix} = 0$$

上式就是一个方程组，有12个未知数。我们可以用6个固定点来求解此方程组。当匹配点大于6对时，可以使用SVD等方法对超定方程求最小二乘解。这种方法需要注意：**求出的R不一定满足R∈SO(3)的要求，因此我们需要寻找一个最好的旋转矩阵对它进行近似**。这可以由QR分解完成。

7.7.2 P3P

这种方法可以只使用三对匹配点。P3P问题如下图所示：

7.7.3 Bundle Adjustment

除了使用线性方法外，我们可以把PnP问题构建成一个定义于李代数上的非线性最小二乘问题。这个Bundle Adjustment问题，是一个最小化**重投影误差(Reprojection error)**的问题。

考虑n个三维空间点P和它们的投影p，我们希望计算相机的位姿R，t，根据相机模型：

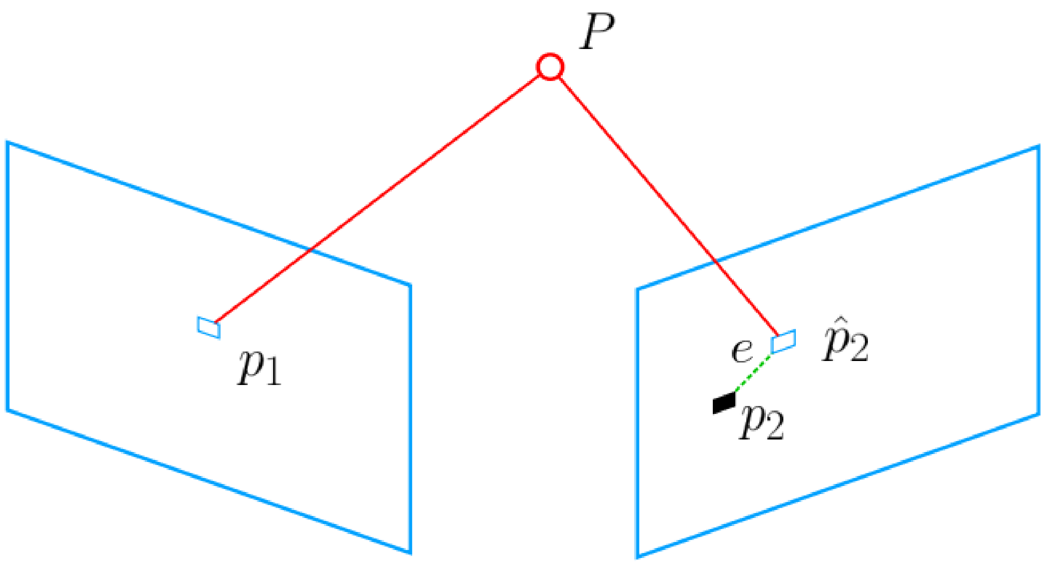
$$s_i \begin{bmatrix} u_1 \\ v_i \\ 1 \end{bmatrix} = Kexp(\hat{\xi}) \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

其中，空间点坐标为Pi=[X-,Yi,Zi]，像素坐标为ui=[ui,vi]^T。我们直接求其重投影误差，再进行最小二乘优化：

$$\xi^* = argmin_{\xi}\frac{1}{2}\sum_{i=1}^{n}||u_i - \frac{1}{s_i}Kexp(\hat{\xi})P_i||^2$$

该问题的误差项就是**重投影误差**。*此处涉及到非齐次坐标转齐次坐标的过程。*

这个最小二乘问题有3维(因为ui是3维，后面齐次转为非齐次也变为3维)，但是ui的最后一维必等于1，因此最后一维误差为零。因此问题变为2维。这个过程中：由于相机位姿变化，同一个点的投影变化，两点间产生距离，因此我们调整相机位姿使得这个距离变小，如下图所示：



计算误差e的变化关于扰动量的导数，根据链式法则：

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

这里的 ◯ + 是指李代数上的左乘扰动。对e求P'的偏导数易得：

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

第二项为变换后的点关于李代数的导数：

$$\frac{\partial \boldsymbol{P}'}{\partial \delta \boldsymbol{\xi}} = [\boldsymbol{I}, -\boldsymbol{P}'^{\wedge}].$$

我们取出上式的前三行，进行乘积，得到：

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

这样我们就得到了误差e关于李代数的导数。

7.8 3D-3D：ICP

3D-3D的位姿估计问题：假设我们有一组配对好的3D点(比如我们对两个RGB-D图像进行了匹配)：

$$P = \{p_1, \ldots, p_n\}, \; P' = \{p'_1, \ldots, p'_n\}$$

现在，找一个欧式变换R，t，使得：

$$p_i = Rp_i' + t$$

这个问题可以用迭代最近点(Iterative Closest Point , ICP)进行求解。

7.8.1 SVD方法

构建最小二乘问题：

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

下面推导其求解方法：首先，定义两组点的质心：

随后，在误差函数中，我们作如下的处理：

注意到：交叉项部分中，(pi-p-R(pi'-p))在求和之后是为零的，因此优化目标函数可以简化为：

7.8.2 非线性优化方法