

三维计算机视觉的部分数学原理

王啸峰 from UCAS & CASIA

2021 年 6 月 24 日

1 相机模型与多视几何

1.1 欧式空间和射影空间

先说故事：三维欧式空间可以很好的描述我们三维世界的“坐标点”，但是一旦涉及到二维图像，三维欧式坐标就不太好描述图像上像素点的位置了（因为图像上一个像素点对应的是三维空间中的一条直线：光心和像素点连成的直线），所以需要射影空间来帮我们描述二维图像上的点和我们三维空间坐标的关系。以后我们就可以利用射影空间的性质（这部分比较复杂，本文只是粗略涉及）方便地计算出我们想要的空间和图像上的几何关系。

1.1.1 二维射影空间 = 变换尺度因子 λ \times 齐次化的二维图像坐标 = 三维欧式空间过原点的直线 + 无穷远球面？

小标题的含义只是粗略的理解这两个空间的变换关系（并且这个二维射影空间可能的表述应该是三维：多加了一维是 λ ），下面给出数学表达：

我们看下面图片， $[x, y]^T$ 是二维图像平面上的坐标， $[X, Y, Z]$ 是欧式空间中的三维坐标， f 是焦距，根据红绿两个相似三角形，我们可以得到：

$$\lambda \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (1)$$

其中 λ 就是尺度变换因子。当 λ 变换的时候，描述的都是该像素点对应的三维空间直线上的所有点。所以 λ 的变化并不影响「像素点和三维直线的关系」，所以我们将上公式简化为：

$$\lambda \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \lambda f \begin{bmatrix} \frac{x}{f} \\ \frac{y}{f} \\ 1 \end{bmatrix} = \lambda_{new} \begin{bmatrix} x_{new} \\ y_{new} \\ 1 \end{bmatrix} \quad (2)$$

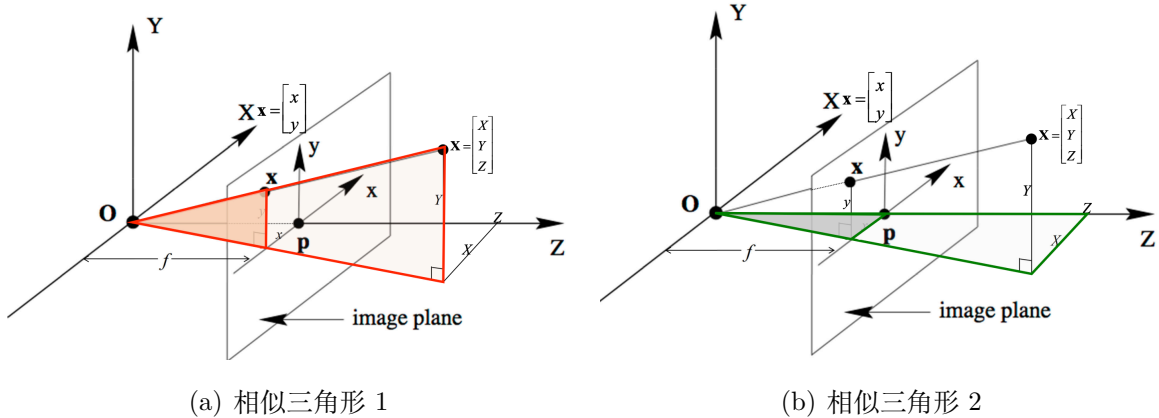


图 1: 射影空间和欧式空间关系说明

此时的 $[x_{new}, y_{new}]^T$ 可以看作焦距为 1 时的图像平面上的点，但是我们将它齐次化之后，得到了 $[x_{new}, y_{new}, 1]^T$ ，他就有了新的含义，在这里我把它理解为二维射影空间里的点。在这里我们可以看出，二维射影空间乘以一个尺度变换因子 λ_{new} 就可以表示出三维欧式空间 (过原点) 的一条直线，即把二维平面和三维空间联系在了一起。(换一个角度思考，三维空间过原点的一条直线可以用三个参数来描述这个直线，也就是 $\lambda_{new}, x_{new}, y_{new}$)。

其实这种表示方法还不止可以表示出三维欧式空间过原点的所有直线，它还可以表示出三维欧式空间的无穷远点的集合 (将它定义成一个球面)。这种表示方法为：

$$\lambda \begin{bmatrix} x \\ y \\ f \end{bmatrix} = \lim_{F \rightarrow +\infty} \lambda F \begin{bmatrix} \frac{x}{F} \\ \frac{y}{F} \\ \frac{f}{F} \end{bmatrix} = \lambda_{new} \begin{bmatrix} x_{new} \\ y_{new} \\ 0 \end{bmatrix} \quad (3)$$

至于为啥要这么定义和表示，这样可以满足很多数学性质，以及很好的处理显示生活中平行但是图像上不平行的两条直线 (例如消失在视野中的两条铁轨)。具体的性质可以见 <https://blog.csdn.net/frozenspring/article/details/76034356?spm=1001.2014.3001.5501>。

综上所述，我们可以得到最一般的射影空间 (可理解为焦距为 1 情况下) 表述如下 (把 λ 去掉是因为等式左右两边在相差一个尺度因子的情况下是等价的)：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (4)$$

$$\begin{bmatrix} x \\ y \\ 0 \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} \infty \\ \infty \\ \infty \end{bmatrix} \quad (5)$$

1.2 相机模型

描述相机的内外参数，内指的是焦距等参数，描述的是以相机为原点看世界；外指的是从相机原点到世界原点的旋转和平移关系。

最终得到的点就是二维像素点和世界三维欧式坐标点的关系，所以要借助上面的射影空间。

1.2.1 相机内参数

上面 4,5描述的是最一般射影空间 (焦距为 1), 在实际的相机模型描述中, 我们还保留焦距为 f 的射影空间表示方法:

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} = \lambda \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \quad (6)$$

因为 f 是个未知数, 所以我们要把他从变量中提取出来:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} 1 & & \\ & 1 & \\ & & \frac{1}{f} \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \stackrel{\lambda}{=} \mathbf{K} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \quad (7)$$

其中的 $\begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix}$ 就是相机的内参数, 记作 \mathbf{K} , f 的单位应该是像素 (焦距物理尺寸/单个 CCD 感光元宽度), 当然这是理想情况下的。实际情况下考虑到感光元不是个标准的正方形 (f_x, f_y 代表长宽, s) 代表夹角, 并且图像的中心不是在 CCD 的中心而是在左上角 (p_x, p_y), 真实的内参是:

$$\begin{bmatrix} f_x & s & p_x \\ & f_y & p_y \\ & & 1 \end{bmatrix}$$

1.2.2 相机外参数

上面描述的是像素点到相机坐标系下的三维坐标点的关系, 但是在真实世界中我们需要知道像素点和世界坐标系下三维坐标的关系, 相机外参数就是来描述相机坐标系到世界坐标系的转换关系。

这个就更简单了, 因为两个三维坐标系的转换只涉及到旋转和平移 (涉及到平移所以

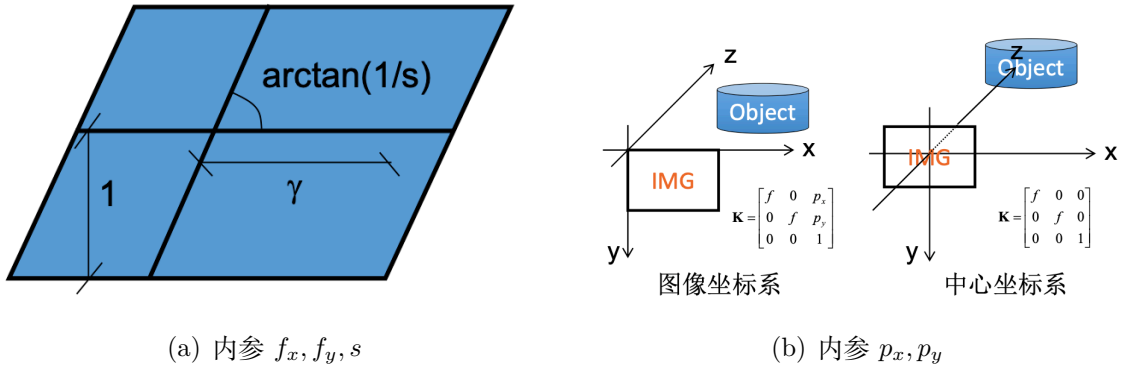


图 2: 相机内参数示意图

要先齐次化),

$$\begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \\ 1 \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{bmatrix} \quad (8)$$

那么这个外参数矩阵与上面公式中的旋转矩阵 \mathbf{R} 和平移向量 \mathbf{t} 有关。

1.2.3 总结

综上所述, 想要把齐次化像素点 (三维) 和齐次化的世界坐标 (四维) 联系起来, 还需要一个 3×4 的转换矩阵, 表述如下:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}^T & 1 \end{bmatrix} \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{bmatrix} \quad (9)$$

简化如下:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \stackrel{\lambda}{=} \mathbf{K} [\mathbf{R} | \mathbf{t}] \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{bmatrix} \quad (10)$$

其中 \mathbf{K} 是相机内参矩阵

$$\begin{bmatrix} f & & \\ & f & \\ & & 1 \end{bmatrix}$$

上述公式都是 $\stackrel{\lambda}{=}$ (差一个尺度因子时等价), 但是这是简化在焦距 $f = 1$ 的 2 维平面上的, 实际上在使用的时候, 例如我们想把 3 维点投射到 2 维焦距为 f 的图像平面上, 我们就不能在 $\stackrel{\lambda}{=}$ 上讨论了, 要严格相等才行。

回忆公式 6:

$$\begin{bmatrix} x \\ y \\ f \end{bmatrix} = \lambda \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix} \stackrel{\lambda}{=} \begin{bmatrix} X_{cam} \\ Y_{cam} \\ Z_{cam} \end{bmatrix}$$

即 $\lambda = \frac{f}{Z_{cam}}$, 将 λ 带入公式 10, 又考虑到公式 7 的第二个 $\stackrel{\lambda}{=}$, 已经乘以过一次 f , 所以最终 $\lambda = \frac{1}{Z_{cam}}$, 最终得到严格相等的相机模型:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \stackrel{\lambda}{=} \mathbf{K} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{bmatrix} = \frac{1}{Z_{cam}} \mathbf{K} [\mathbf{R}|\mathbf{t}] \begin{bmatrix} X_{world} \\ Y_{world} \\ Z_{world} \\ 1 \end{bmatrix} \quad (11)$$

1.3 多视几何

多视几何表述的就是从不同位置去拍摄同一个东西，相当于同一世界坐标系下的物体在不同的照片上有不同的像素位置，根据这些对应关系，我们可以构造线性方程求解相机的内外参数矩阵，甚至是物体在世界坐标系下的坐标。

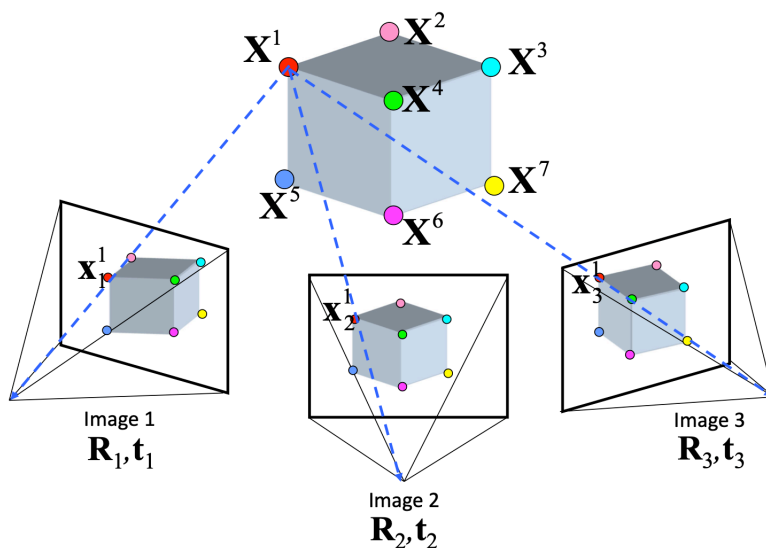


图 3: 多视几何

1.3.1 两视图几何—基本矩阵和本质矩阵

多视角有点复杂，我们先从两视图来分析。这个基本矩阵呢，描述的就是两幅图像上“对应点像素位置”的关系。本质矩阵呢，就是描述两幅图像上“对应点的相机坐标系下的三维坐标”的关系。这俩玩意其实道理是一样的，不过不同的地方用不同的表述方法可能方便点。

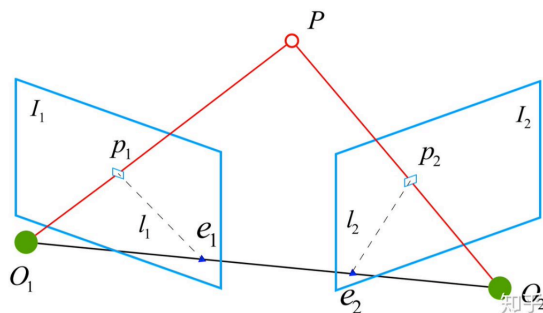


图 4: 两视图对极几何

如图 4 所示, PO_1O_2 是极平面; O_1O_2 是基线; e_1 和 e_2 是对极点; p_1e_1 和 p_2e_2 是对极线。

想求得我们开始说的点的关系, 就要用到 O_1P, O_1O_2, O_2P 在同一个平面上这个关系。即三个向量的叉乘为 0。即 $O_1P \cdot (O_1O_2 \times O_2P) = 0$

那么 O_1P, O_1O_2, O_2P 分别是什么呢? 我们可以分析一下: 假设 P 在 O_1 为原点的相机坐标系下点坐标是 \mathbf{X}_1 , 在 O_2 为原点的相机坐标系下点坐标是 \mathbf{X}_2 , 且 O_1 坐标系和 O_2 的关系是: O_1 先平移 \mathbf{t} , 然后再逆时针旋转 θ 矩阵 $\mathbf{R} = \begin{bmatrix} \cos(\theta) & \sin(-\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$, 那么点 \mathbf{X}_1 和点 \mathbf{X}_2 的关系是: \mathbf{X}_2 先旋转 \mathbf{R} , 在平移 \mathbf{t} 得 \mathbf{X}_1 (坐标系的变化和坐标点的变化是反过来的, 见附录推导)。所以 $\mathbf{X}_1 = \mathbf{R}\mathbf{X}_2 + \mathbf{t}$ 。

以 O_1 为原点建立相机坐标系, 我们可以得到:

$$O_1O_2 = \mathbf{t},$$

$$O_1P = \mathbf{X}_1,$$

$$O_2P = \mathbf{X}_1 - \mathbf{t} = \mathbf{R}\mathbf{X}_2.$$

进一步计算 $O_1P \cdot (O_1O_2 \times O_2P) = 0$, 有: $\mathbf{X}_1^T(\mathbf{t} \times \mathbf{R}\mathbf{X}_2) = 0$, 这个叉乘我们可以表示成矩阵的形式 (转换关系见附录):

$$\mathbf{X}_1^T([\mathbf{t}]_{\times}\mathbf{R})\mathbf{X}_2 = 0 \quad (12)$$

这里的 $[\mathbf{t}]_{\times}\mathbf{R}$ 就是本质矩阵 \mathbf{E} 啦!!!! 描述的是两个相机坐标系下 P 点坐标的转换关系。

下面的基本矩阵是一个道理, 不过描述的是两张图片上像素位置 (p_1 和 p_2) 的转换关系。根据公式 7: $\mathbf{x} \stackrel{\lambda}{=} \mathbf{K}\mathbf{X}$, 有:

$$\mathbf{x}_1^T(\mathbf{K}_1^{-T}[\mathbf{t}]_{\times}\mathbf{R}\mathbf{K}_2^{-1})\mathbf{x}_2 = 0 \quad (13)$$

其中 $\mathbf{K}_1^{-T}[\mathbf{t}]_{\times}\mathbf{R}\mathbf{K}_2^{-1}$ 就是基础矩阵 \mathbf{F} 。

1.3.2 两视图几何-8 点法

根据基础矩阵 \mathbf{F} 和给定两视图中的对应点 x, x' , 我们有 $\mathbf{x}^T\mathbf{F}\mathbf{x} = 0$, 按我们的第一印象, 矩阵含有两个内参矩阵 (10DoF), 一个旋转矩阵 (3DoF), 一个平移矩阵 (3DoF), 一

共有 16 个未知数，但是要知道这些未知数不是独立的，我们展开 $\mathbf{x}'^T \mathbf{F} \mathbf{x} = 0$ 可以得到：

$$[x_1 x_2, x_1 y_2, x_1, y_1 x_2, y_1 y_2, y_1, x_2, y_2, 1] \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \quad (14)$$

所以实际的未知参数是 9，但是最后一个维度的变量是 1，所以实际未知参数是 8 个，即 8 个方程求解：

$$\begin{bmatrix} x_1^1 x_2^1 & x_1^1 y_2^1 & x_1^1 & y_1^1 x_2^1 & y_1^1 y_2^1 & y_1^1 & x_2^1 & y_2^1 & 1 \\ x_1^2 x_2^2 & x_1^2 y_2^2 & x_1^2 & y_1^2 x_2^2 & y_1^2 y_2^2 & y_1^2 & x_2^2 & y_2^2 & 1 \\ x_1^3 x_2^3 & x_1^3 y_2^3 & x_1^3 & y_1^3 x_2^3 & y_1^3 y_2^3 & y_1^3 & x_2^3 & y_2^3 & 1 \\ x_1^4 x_2^4 & x_1^4 y_2^4 & x_1^4 & y_1^4 x_2^4 & y_1^4 y_2^4 & y_1^4 & x_2^4 & y_2^4 & 1 \\ x_1^5 x_2^5 & x_1^5 y_2^5 & x_1^5 & y_1^5 x_2^5 & y_1^5 y_2^5 & y_1^5 & x_2^5 & y_2^5 & 1 \\ x_1^6 x_2^6 & x_1^6 y_2^6 & x_1^6 & y_1^6 x_2^6 & y_1^6 y_2^6 & y_1^6 & x_2^6 & y_2^6 & 1 \\ x_1^7 x_2^7 & x_1^7 y_2^7 & x_1^7 & y_1^7 x_2^7 & y_1^7 y_2^7 & y_1^7 & x_2^7 & y_2^7 & 1 \\ x_1^8 x_2^8 & x_1^8 y_2^8 & x_1^8 & y_1^8 x_2^8 & y_1^8 y_2^8 & y_1^8 & x_2^8 & y_2^8 & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{12} \\ f_{13} \\ f_{21} \\ f_{22} \\ f_{23} \\ f_{31} \\ f_{32} \\ f_{33} \end{bmatrix} = 0 \quad (15)$$

2 相机标定与稀疏重建

首先说一下多视几何引发出来的几个问题, 假设我得到了多视几何上 n 个点:

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{K}_1 [\mathbf{R}_1 \mid \mathbf{t}_1] \mathbf{X} \\ &\vdots \\ \mathbf{x}_n &= \mathbf{K}_n [\mathbf{R}_n \mid \mathbf{t}_n] \mathbf{X} \end{aligned}$$

- 三角化: 已知 $\mathbf{x}, \mathbf{K}, \mathbf{R}, \mathbf{t}$, 求 \mathbf{X}
- 姿态估计: 已知 $\mathbf{x}, \mathbf{X}, \mathbf{K}$, 求 \mathbf{R}, \mathbf{t}
- 相机标定: 已知 \mathbf{x}, \mathbf{X} , 求 $\mathbf{K}, \mathbf{R}, \mathbf{t}$
- 稀疏重建: 已知 \mathbf{x} , 求 $\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{X}$

他们的求解有共同之处, 其实就是列出线性方程, 求解位置参数 (由上到下需要的方程数要变多), 但是由于误差的存在肯定无解, 只能用伪逆求解线性方程, 然后得到初始解后在逐步迭代优化这个解。

下面看看相机标定和稀疏重建这两个方面。

2.1 相机标定

下面就是最一般的情况 $p_{ij} \in (3, 4)$ 是未知的。也是我们要求解的。

$$\begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_i \\ Y_i \\ Z_i \\ 1 \end{bmatrix}$$

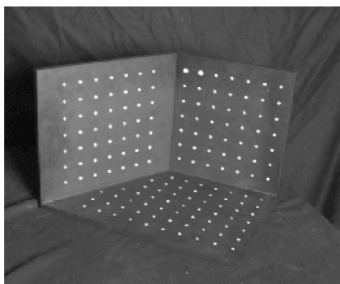


图 5: 最一般的相机标定方式

我们将其展开，可以得到：

$$\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -x_i X_i & -x_i Y_i & -x_i Z_i & -x_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -y_i X & -y_i Y & -y_i Z_i & -y_i \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{14} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{24} \\ p_{31} \\ p_{32} \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

即一共 12 个未知参数（11Dof「有时间在附录解释下」），但是一副对应点可以得到两个线性方程，所以最少只需要 6 个点得 12 个线性方程即可求解。但是这种方法是冗余的，我们可以参考下一种标定方法：

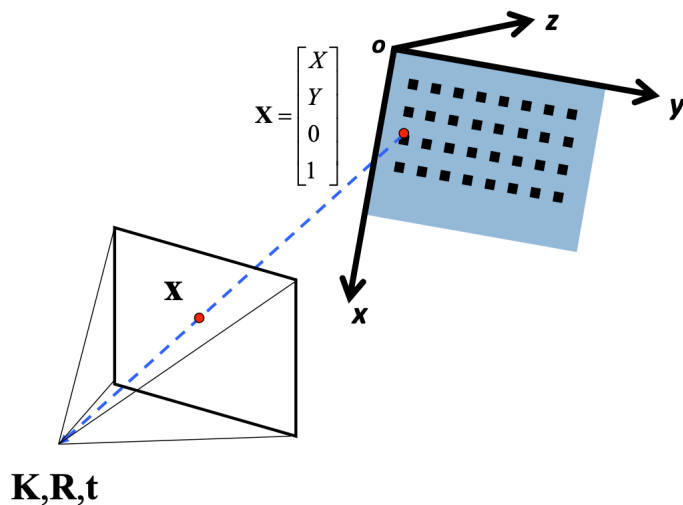


图 6: 张友正标定法

通过限制三维世界坐标的 $Z = 0$ ，我们可以简化方程为：

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}$$

其中 $\mathbf{K} \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix}$ 为单应矩阵 $\mathbf{H} \in (3, 3)$ (DOF=8), 此时我们只需要 4 个对应点就得到 8 个方程就可以求解出相机参数。

到现在为止, 我们求的是 \mathbf{H} 而不是相机内外参数 $\mathbf{K}, \mathbf{R}, \mathbf{t}$, 那么他们的转换关系是啥呢? 先把上述公式转换为:

$$\begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t} \end{bmatrix} = \begin{bmatrix} \mathbf{K}^{-1}\mathbf{h}_1 & \mathbf{K}^{-1}\mathbf{h}_2 & \mathbf{K}^{-1}\mathbf{h}_3 \end{bmatrix} \quad (16)$$

再考虑到旋转向量的两个性质: $\mathbf{r}_1^T \mathbf{r}_2 = 0$ $\|\mathbf{r}_1\| = \|\mathbf{r}_2\| = 1$

所以得到两个约束方程:

$$\begin{aligned} \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 &= 0 \\ \mathbf{h}_1^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_1 &= \mathbf{h}_2^T \mathbf{K}^{-T} \mathbf{K}^{-1} \mathbf{h}_2 \end{aligned}$$

定义 $\boldsymbol{\omega} = \mathbf{K}^{-T} \mathbf{K}^{-1}$, 我们通过 3 组 h 即 6 个方程就可以求解出 $\boldsymbol{\omega}$ (因为 \mathbf{K} 的 DoF=5), 求出 $\boldsymbol{\omega}$ 后再正交分解 (QR 分解: 正交 + 上三角) 求出 \mathbf{K} , 然后根据公式 16 直接求解 r_1, r_2, t , 然后 $r_3 = r_1 \times r_2$

2.2 姿态估计

PnP 就是这个里面的问题, 这里的 n 就是给定的匹配点的个数。然后再给定相机内参 \mathbf{K} , 要求外参数 \mathbf{R}, \mathbf{t} .

- P2P: 无穷多解;
- P3P: 最多四个解, 需要第 4 个点验证;
- P4P: 最多四个解, 当四点共面时有唯一解;
- P5P: 最多两个解;
- EPnP: 4 组不共面点时有唯一解;
- PnP: 线性求解 ($n > 5$).

2.3 稀疏重建-BA-SFM

我们回头看基本矩阵 $\mathbf{x}_1^T \mathbf{F} \mathbf{x}_2 = 0$, 仿佛就是来解决这个问题的, 按照“两视图几何-8点法”可以解出 \mathbf{F} , 不幸的是, \mathbf{F} 无法唯一分解出 $\mathbf{K}_1, \mathbf{K}_2, \mathbf{R}, \mathbf{t}$.

但是我们放松一下条件, 如果假设 $\mathbf{K}_1, \mathbf{K}_2$ 已知 (这个也比较好获得, 直接相机说明书等获得)。这样我们可以把目光投降本质矩阵 $\mathbf{X}_1^T \mathbf{E} \mathbf{x}_2 = 0$.

那么我们可以先通过 5 点法求得 \mathbf{E} , 然后通过 SVD 分解求解 \mathbf{R}, \mathbf{t} 。下面我们不加证明直接给出解:

假设相机 1 的 $\mathbf{R}_1 = \mathbf{I}, \mathbf{t}_1 = \mathbf{0}$, SVD 分解得 $\mathbf{E} = \mathbf{U} \text{diag}(1, 1, 0) \mathbf{V}^T$ 。此时相机 2 有 4 组解:

$$\begin{aligned} [\mathbf{R}_2 | \mathbf{t}_2] &= [\mathbf{U} \mathbf{W} \mathbf{V}^T | +\mathbf{u}_3] \\ [\mathbf{R}_2 | \mathbf{t}_2] &= [\mathbf{U} \mathbf{W} \mathbf{V}^T | -\mathbf{u}_3] \\ [\mathbf{R}_2 | \mathbf{t}_2] &= [\mathbf{U} \mathbf{W}^T \mathbf{V}^T | +\mathbf{u}_3] \\ [\mathbf{R}_2 | \mathbf{t}_2] &= [\mathbf{U} \mathbf{W}^T \mathbf{V}^T | -\mathbf{u}_3] \end{aligned}$$

其中 $\mathbf{W} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

接下来怎么从这四个解中选一个呢? 我们的原则是通过「三角化」求解 \mathbf{X} , 然后保留 \mathbf{X} 在相机前方最多的那个解。

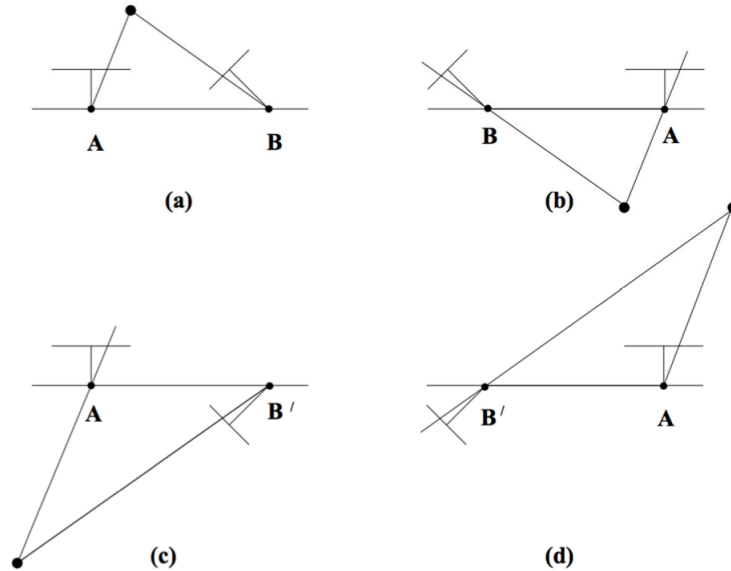


图 7: 稀疏重建解 4 选 1

这个可以当作初始解 (代数误差最小), 下面我们要迭代获得一个更精确的解。即求解

重投影误差最小化（几何误差最小）：

描述问题如下：在给定 \mathbf{K} ，通过对 \mathbf{E} 进行 SVD 分解后求得 \mathbf{R}, \mathbf{t} ，并三角化得到 \mathbf{X} ，迭代求解重投影误差最小化问题：

$$\min \sum_i \|\mathbf{P}\mathbf{X}_i - \mathbf{x}_i\|^2$$

最终得到几何误差最小化的 $\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{X}$ 。

如本节开头所说，其实三角化，相机标定，稀疏重建都可以先线性求一个初始解，然后在迭代求解重投影误差最小化，他们的异同如下图：

三角化： $\min \sum_i \|\mathbf{P}_i \mathbf{X} - \mathbf{x}_i\|^2$ $\mathbf{x}, \mathbf{K}, \mathbf{R}, \mathbf{t}$ 已知，求 \mathbf{X}

相机标定： $\min \sum_i \|\mathbf{P} \mathbf{X}_i - \mathbf{x}_i\|^2$ \mathbf{x}, \mathbf{X} 已知，求 $\mathbf{K}, \mathbf{R}, \mathbf{t}$

稀疏重建： $\min \sum_i \|\mathbf{P}_i \mathbf{X}_i - \mathbf{x}_i\|^2$ \mathbf{x} 已知，求 $\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{X}$

图 8: 重投影误差最小化

下面长话短说，上面的 \mathbf{X}_i 在不同问题中含义不同，但是是 $\mathbf{K}, \mathbf{R}, \mathbf{t}, \mathbf{X}$ 的非线性组合，所以这可以看作非线性的最小二乘优化问题。求解非线性最小二乘一般会用 LM 法（一种启发式的阻尼高斯牛顿法）。

所谓的 BA: Bundle Adjustment 捆绑调整，说的就是用 LM 求解重投影误差最小化。

上面这一套稀疏重建从最小化代数误差然后再到 BA 优化得到最小化几何误差就是 SFM(structure from motion)。一般比较好做的是二视图的 SFM，增量式的 SFM 流程如下：

- 选择初始图像对；
- 通过两视图 SfM 计算初始模型；
- 如果图像连接关系图还有未选择的图像
 - 选择一幅能看目前模型中最多 3D 点的图像；
 - 根据 3D-2D 点对应估计相机位姿；
 - 三角化新的特征点 Tracks
 - bundle adjustment

3 附录

3.1 坐标变换和坐标轴变换的关系

如图 9所示:

O_1 先向右上角平移一格, 再逆时针旋转 45 度, 得 O_2 , 那么对于坐标系上的点来说, 是相反的。

P_2 先逆时针旋转 45 度, 再右上移动一格, 得 P_1 . 证明如下:

$$P_1 = \begin{bmatrix} 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(45) & \sin(-45) & 1 \\ \sin(45) & \cos(45) & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{2} \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(-\theta) & t_x \\ \sin(\theta) & \cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} P_2$$

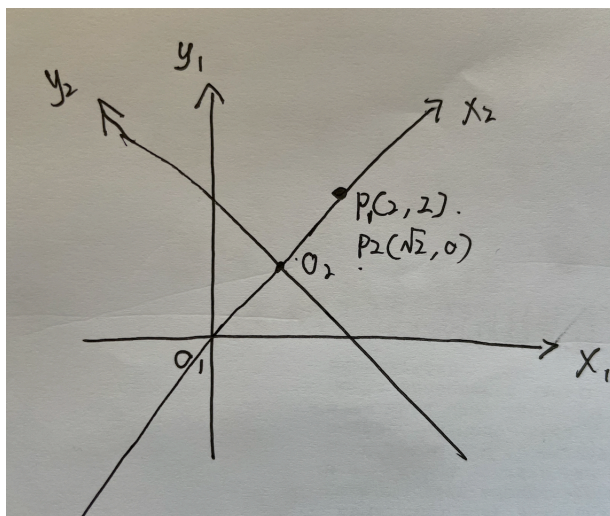


图 9: 旋转平移变换

3.2 向量叉乘转矩阵形式

两个向量如下:

$$a = (a_1, a_2, a_3)$$

$$b = (b_1, b_2, b_3)$$

通过引入单位向量, 向量就可以转化为代数形式:

$$a = a_1i + a_2j + a_3k$$

$$b = b_1i + b_2j + b_3k$$

定义单位向量间的运算规则

$$\begin{aligned}i * i &= 0 & j * j &= 0 & k * k &= 0 \\i * j &= k & j * k &= i & k * i &= j \\j * i &= -k & k * j &= -i & i * k &= -j\end{aligned}$$

计算叉乘：

$$\begin{aligned}a \times b &= (a_1 i + a_2 j + a_3 k) * (b_1 i + b_2 j + b_3 k) \\a \times b &= (a_2 b_3 - a_3 b_2) i + (a_3 b_1 - a_1 b_3) j + (a_1 b_2 - a_2 b_1) k\end{aligned}$$

写成向量的形式：

$$a \times b = \begin{bmatrix} a_2 b_3 - a_3 b_2 \\ a_3 b_1 - a_1 b_3 \\ a_1 b_2 - a_2 b_1 \end{bmatrix}$$

变换得到叉乘矩阵：

$$a \times b = [a]_{\times} b = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$