# Ch2 Image formation

## 2.1 Geometric primitives and transformations

首先介绍一些用于构建三维物体的几何基元：点、线和面
**2D 点**
2D 点，也就是图像中的像素坐标，可以用一对值来表示：$\boldsymbol{x} = (x, y) \in \mathcal{R}^2$。
也可以通过齐次坐标系 (homogeneous coordinates) 表示：$\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) \in \mathcal{P}^2$，$\mathcal{P}^2 = \mathcal{R}^3 - (0, 0, 0)$ 称为 2D 投影空间。
通过除以最后一个元素 $\tilde{w}$ 可以将齐次向量 $\tilde{\boldsymbol{x}}$ 转换回非齐次向量 $\boldsymbol{x}$：

$$\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{w}) = \tilde{w}(x, y, 1) = \tilde{w}\bar{\boldsymbol{x}}$$

其中 $\bar{\boldsymbol{x}}$ 称为增广向量 (augmented vector)。在齐次坐标系中最后一个元素 $\tilde{w} = 0$ 的点被称为理想点 (ideal points) 或无穷远点 (points at infinity)，其实就对应了笛卡尔坐标系中的 $(\infty, \infty)$。
**2D 直线**
2D 直线也可以通过齐次坐标表示：$\tilde{\boldsymbol{I}} = (a, b, c)$，对应的直线方程为：

$$\bar{\boldsymbol{x}} \cdot \tilde{\boldsymbol{I}} = ax + by + c = 0$$

我们可以对直线方程向量做归一化，这样就有 $\boldsymbol{l} = (\hat{n}_x, \hat{n}_y, d) = (\hat{\boldsymbol{n}}, d)$，其中 $||\hat{\boldsymbol{n}}|| = 1$，这样的话，$\hat{\boldsymbol{n}}$ 就是这条线的法向量，$d$ 是线到原点的距离。唯一一条不能做归一化的线是位于无穷远处的线 $\tilde{\boldsymbol{I}} = (0, 0, 1)$，它包含了所有的无穷远点。

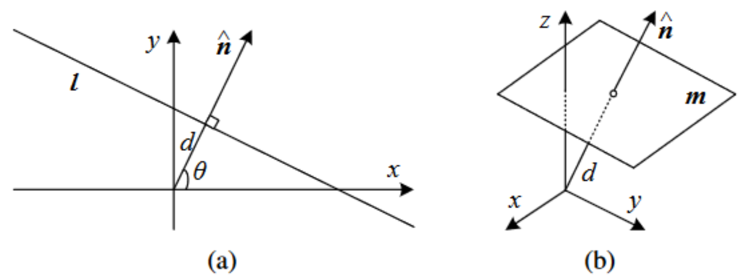



**Figure 2.2** *(a) 2D line equation and (b) 3D plane equation, expressed in terms of the normal n̂ and distance to the origin d.*

在使用齐次坐标系的情况下，我们可以通过叉乘计算两条直线的交点：

$$\tilde{\boldsymbol{x}} = \tilde{\boldsymbol{l}}_1 \times \tilde{\boldsymbol{l}}_2$$

2 个向量叉乘的结果一定会垂直于这两个向量，因此对于这两条直线，都有：

$$(\tilde{\boldsymbol{l}}_1 \times \tilde{\boldsymbol{l}}_2) \cdot \tilde{\boldsymbol{I}} = 0$$

这说明叉乘的结果既在直线 $\tilde{\boldsymbol{I}}_1$ 上又在直线 $\tilde{\boldsymbol{I}}_2$ 上，所以就是两直线的交点。
同理，经过两个点的直线可以写成：

$$\tilde{\boldsymbol{I}} = \tilde{\boldsymbol{x}}_1 \times \tilde{\boldsymbol{x}}_2$$

**3D 点**
与 2D 点同理，可以用齐次坐标系写为 $\tilde{\boldsymbol{x}} = (\tilde{x}, \tilde{y}, \tilde{z}, \tilde{w}) \in \mathcal{P}^3$。
**3D 平面**
3D 平面同样可以通过齐次坐标表示：$\tilde{\boldsymbol{m}} = (a, b, c, d)$，对应的平面方程为：

$$\bar{\boldsymbol{x}} \cdot \tilde{\boldsymbol{m}} = ax + by + cz + d = 0$$

同样可以对平面方程向量做归一化，这样就有 $\boldsymbol{m} = (\hat{n}_x, \hat{n}_y, \hat{n}_z, d) = (\hat{\boldsymbol{n}}, d)$，其中 $||\hat{\boldsymbol{n}}|| = 1$，这样的话，$\hat{\boldsymbol{n}}$ 就是这个面的法向量，$d$ 是原点到平面的距离。位于无穷远处的平面可以表示为 $\tilde{\boldsymbol{m}} = (0, 0, 0, 1)$
**3D 直线**
可以用端点表示法来表示 3D 直线，直线上的任意一点可以表示为两个点的线性组合：

$$\boldsymbol{r} = (1 - \lambda)\boldsymbol{p} + \lambda \boldsymbol{q}$$

特殊情况是第二个端点是无穷远点，即 $\tilde{\boldsymbol{q}} = (\hat{d}_x, \hat{d}_y, \hat{d}_z, 0) = (\hat{\boldsymbol{d}}, 0)$，可以看出 $\hat{\boldsymbol{d}}$ 其实是直线的方向，因此可以写为：

$$\boldsymbol{r} = \boldsymbol{p} + \lambda \hat{\boldsymbol{d}}$$

端点表示法的缺点在于自由度太多，每个点都需要 3 个自由度，但一条直线实际上只需要 4 个自由度。一种解决方法就是固定 2 个平面，在平面中去取端点。例如固定 $z = 0$ 和 $z = 1$ 这两个平面，在两个平面中分别取点作为端点，这样只需要确定 $(x, y)$，此时自由度为 4。

## 2.1.1 2D transformations

| Transformation | Matrix | # DoF | Preserves | Icon |
|---|---|---|---|---|
| translation | $\begin{bmatrix} \mathbf{I} & \mathbf{t} \end{bmatrix}_{2\times3}$ | 2 | orientation | |
| rigid (Euclidean) | $\begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix}_{2\times3}$ | 3 | lengths | |
| similarity | $\begin{bmatrix} s\mathbf{R} & \mathbf{t} \end{bmatrix}_{2\times3}$ | 4 | angles | |
| affine | $\begin{bmatrix} \mathbf{A} \end{bmatrix}_{2\times3}$ | 6 | parallelism | |
| projective | $\begin{bmatrix} \tilde{\mathbf{H}} \end{bmatrix}_{3\times3}$ | 8 | straight lines | |

**平移 (Translation)**
2D 平移可写为：

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}}$$

**旋转 + 平移**
这种变换也被称为 2D 刚体变换或 2D 欧几里德变换 (因为欧几里德距离得以保留)。可以写成：

$$\boldsymbol{x}' = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}}$$

其中

$$\boldsymbol{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$\boldsymbol{R}$ 是正交矩阵，且 $|\boldsymbol{R}| = 1$。
**Scaled rotation**
也被称为相似变换 (similarity transform)，可以写为：

$$\boldsymbol{x}' = \begin{bmatrix} s\boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \bar{\boldsymbol{x}} = \begin{bmatrix} a & -b & t_x \\ b & a & t_y \end{bmatrix} \bar{\boldsymbol{x}}$$

**仿射 (Affine)**
可以写为：

$$\boldsymbol{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{\boldsymbol{x}}$$

仿射变换保留平行的性质。
**投影 (Projective)**
也可以称为单应性 (homography) 变换，是在齐次坐标系中进行的：

$$\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{H}} \tilde{\boldsymbol{x}}$$

$\tilde{\boldsymbol{H}}$ 是齐次的，两个成比例的矩阵是等价的。得到的齐次坐标系下的 $\tilde{\boldsymbol{x}}'$ 可以通过归一化来得到非齐次的结果 $\boldsymbol{x}$ 即

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}} \text{ and } y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$

**Co-vectors**
上述的变换可以用于对 2D 平面中的点，是否也可以用于直接变换直线方程？考虑齐次坐标下的直线方程 $\tilde{\boldsymbol{l}} \cdot \tilde{\boldsymbol{x}} = 0$，如果我们对直线上的点做变换 $\tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{H}} \tilde{\boldsymbol{x}}$，可以得到：

$$\tilde{\boldsymbol{l}}' \cdot \tilde{\boldsymbol{x}}' = \tilde{\boldsymbol{l}}'^T \tilde{\boldsymbol{H}} \tilde{\boldsymbol{x}} = (\tilde{\boldsymbol{H}}^T \tilde{\boldsymbol{l}}')^T \tilde{\boldsymbol{x}} = \tilde{\boldsymbol{l}} \cdot \tilde{\boldsymbol{x}} = 0$$

注意上式第一个等号，为什么会多一个转置，是因为是从点乘变为了矩阵乘法，所以多个转置符号。
也就是说，对点做变换的结果是 $\tilde{\boldsymbol{l}}' = \tilde{\boldsymbol{H}}^{-T} \tilde{\boldsymbol{l}}$。对如 2D 直线，3D 法线这样的 co-vectors 做投影变换的操作，可以表示成矩阵逆的转置，等价于 $\tilde{\boldsymbol{H}}$ 的伴随变换 (adjoint) 或共轭变换 (应该不是指伴随矩阵 $\tilde{\boldsymbol{H}}^*$ )。

### 2.1.2 3D transformations

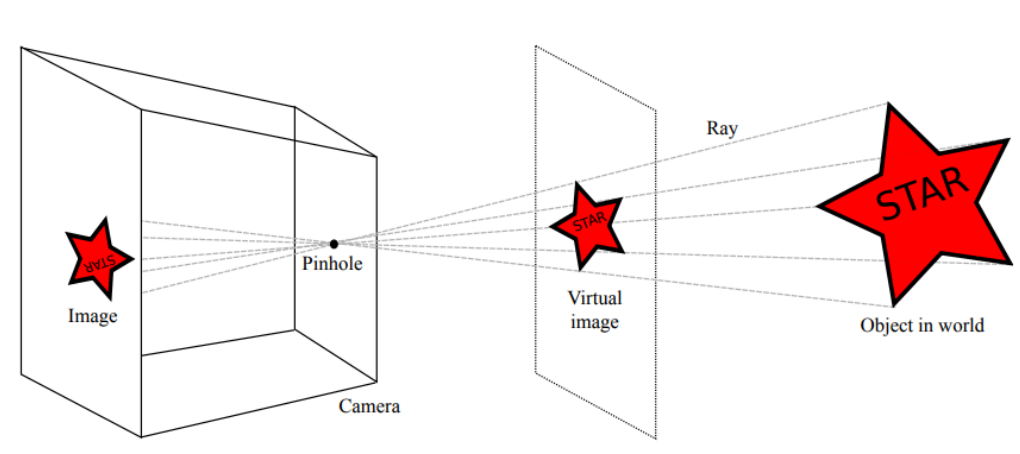
与 2D 变换相似。

| Transformation | Matrix | # DoF | Preserves | Icon |
|---|---|---|---|---|
| translation | $\begin{bmatrix} I & t \end{bmatrix}_{3\times4}$ | 3 | orientation | |
| rigid (Euclidean) | $\begin{bmatrix} R & t \end{bmatrix}_{3\times4}$ | 6 | lengths | |
| similarity | $\begin{bmatrix} sR & t \end{bmatrix}_{3\times4}$ | 7 | angles | |
| affine | $\begin{bmatrix} A \end{bmatrix}_{3\times4}$ | 12 | parallelism | |
| projective | $\begin{bmatrix} \tilde{H} \end{bmatrix}_{4\times4}$ | 15 | straight lines | |

## 2.1.3 3D rotations

## 2.1.4 3D to 2D projections

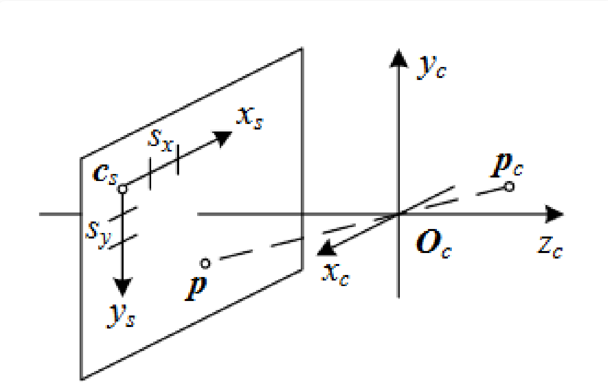
首先介绍针孔相机 (pinhole camera) 模型。来自某个物体的光线穿过相机前部的针孔，在后平面上形成图像。此图像是倒置的，可以考虑位于真空前的虚拟图像平面，这与真实的图像平面等价，且成像是正的。




定义几个概念：虚拟的图像创建于图像平面 (image plane) 上，这个平面放置在沿光学中心 (也就是针孔) 出发的光轴 (optical axis) 上，光轴与像平面相交的点称为像主点 (principal point)，光学中心到像主点之间的距离就是焦距 (focal length)。

### Camera intrinsics

现在我们想要建立相机坐标系中的点到对应图像平面上的点之间的联系。



对于图像传感器平面，图像像素坐标系的原点通常位于左上角，坐标均为整数 $(x_s, y_s)$ ，表示像素索引。为了将像素点映射到相机坐标系中的 3D 点，首先就需要将整数的像素索引值 $(x_s, y_s)$ 乘上像素间距 $(s_x, s_y)$，还需要知道传感器平面原点与相机中心 $\boldsymbol{O}_c$ 之间的旋转朝向 $\boldsymbol{R}_s$，以及位移 $\boldsymbol{c}_s$。
从图像上的 2D 点 $\bar{\boldsymbol{x}}_s$ 到空间中的 3D 点 $\boldsymbol{p}$ (确切说是一条光线) 的投影关系可写为：

$$\boldsymbol{p} = \begin{bmatrix} \boldsymbol{R}_s & \boldsymbol{c}_s \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \boldsymbol{M}_s \bar{\boldsymbol{x}}_s$$

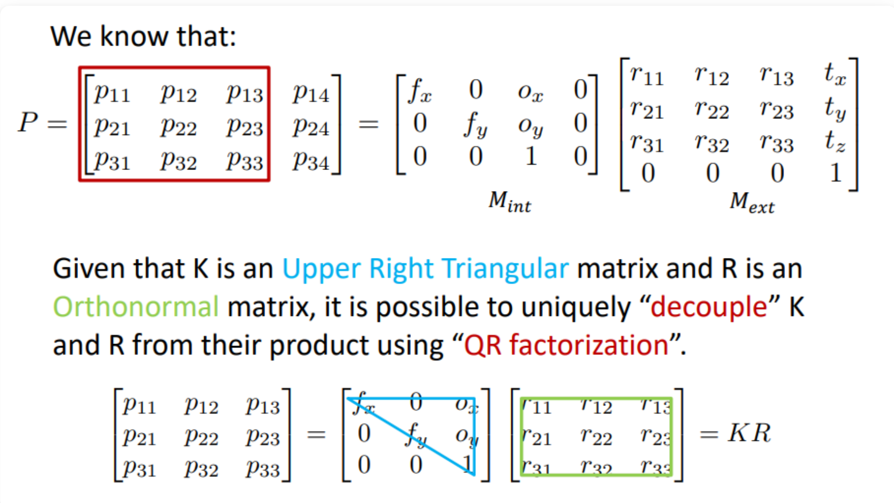
$\boldsymbol{M}_s$ 是一个 $3 \times 3$ 矩阵，前两列表示沿 $x_s$ 和 $y_s$ 方向的步长，第三列表示传感器平面原点 $\boldsymbol{c}_s$。
$\boldsymbol{M}_s$ 有 8 个未知的参数，其中 3 个参数描述旋转 $\boldsymbol{R}_s$，3 个参数描述变换 $\boldsymbol{c}_s$，还有 2 个是缩放系数 $(s_x, s_y)$。但实际上我们只有 7 个自由度，因为通过外部的测量手段，我们无法得知传感器平面与相机中心之间的距离。在实际估计过程中，估计 7 个自由度的参数还是不方便的，因此大部分情况下假设 $\boldsymbol{M}_s$ 是 $3 \times 3$ 的单应矩阵。
由于 $\boldsymbol{p}$ 与 $\boldsymbol{p}_c$ 位于一条光线上，因此可以表示为 $\boldsymbol{p} = s\boldsymbol{p}_c$，因此像素点和空间点之间的关系为：

$$\tilde{\boldsymbol{x}}_s = \alpha \boldsymbol{M}_s^{-1} \boldsymbol{p}_c = \boldsymbol{K} \boldsymbol{p}_c$$

其中 $3 \times 3$ 的 $\boldsymbol{K}$ 就是相机内参矩阵。
根据之前的内容我们知道 $\boldsymbol{K}$ 理论上有 8 个自由度，实际中也有 7 个自由度，但大部分书中都把 $\boldsymbol{K}$ 当作只有 5 个自由度的上三角矩阵，原因就在于通过外部的标定，我们是无法恢复出真实的 $\boldsymbol{K}$。
具体来说，我们是通过一系列的对应点来同时估计相机的内参和外参：

$$\tilde{\boldsymbol{x}}_s = \boldsymbol{K} \begin{bmatrix} \boldsymbol{R} & \boldsymbol{t} \end{bmatrix} \boldsymbol{p}_w = \boldsymbol{P} \boldsymbol{p}_w$$

其中 $\boldsymbol{p}_w$ 是已知的世界坐标系中的 3D 点，$\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$ 是相机矩阵。如果对 $\boldsymbol{K}$ 左乘一个旋转矩阵 $\boldsymbol{R}_1$，对 $[\boldsymbol{R}|\boldsymbol{t}]$ 右乘对应的 $\boldsymbol{R}_1^T$，这仍然是一个有效的标定过程。因此，仅基于外部测量，是无法得到传感器的实际朝向，以及真实的相机内参。
如果规定 $\boldsymbol{K}$ 为上三角矩阵，就能得到合适的相机内外参。通过标定我们得到了 $\boldsymbol{P} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$，我们希望从中恢复出 $\boldsymbol{K}, \boldsymbol{R}$ 和 $\boldsymbol{t}$，可以采用 QR 分解的方法，从 $\boldsymbol{P}$ 的前三列分解得到 $\boldsymbol{K}$ 和 $\boldsymbol{R}$ ($\boldsymbol{K}$ 是上三角矩阵，$\boldsymbol{R}$ 是正交矩阵) ，然后再根据求得的 $\boldsymbol{K}$ 和 $\boldsymbol{P}$ 的第四列反推得到 $\boldsymbol{t}$。



We know that:

$$P=\begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}=\begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

That is:

$$\begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}=\begin{bmatrix} f_x & 0 & o_x \\ 0 & f_y & o_y \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}=K\mathbf{t}$$

Therefore:

$$\boxed{\mathbf{t}=\mathbf{K}^{-1}\begin{bmatrix} p_{14} \\ p_{24} \\ p_{34} \end{bmatrix}}$$

$\boldsymbol{K}$ 可能的形式为：

$$\boldsymbol{K}=\begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

其中 $s$ 表示了由于传感器没有垂直于光轴安装导致传感器两个轴之间的倾斜。$(c_x, c_y)$ 为像素坐标系下的图像中心位置，也就是之前所说的像主点 (principal point)。