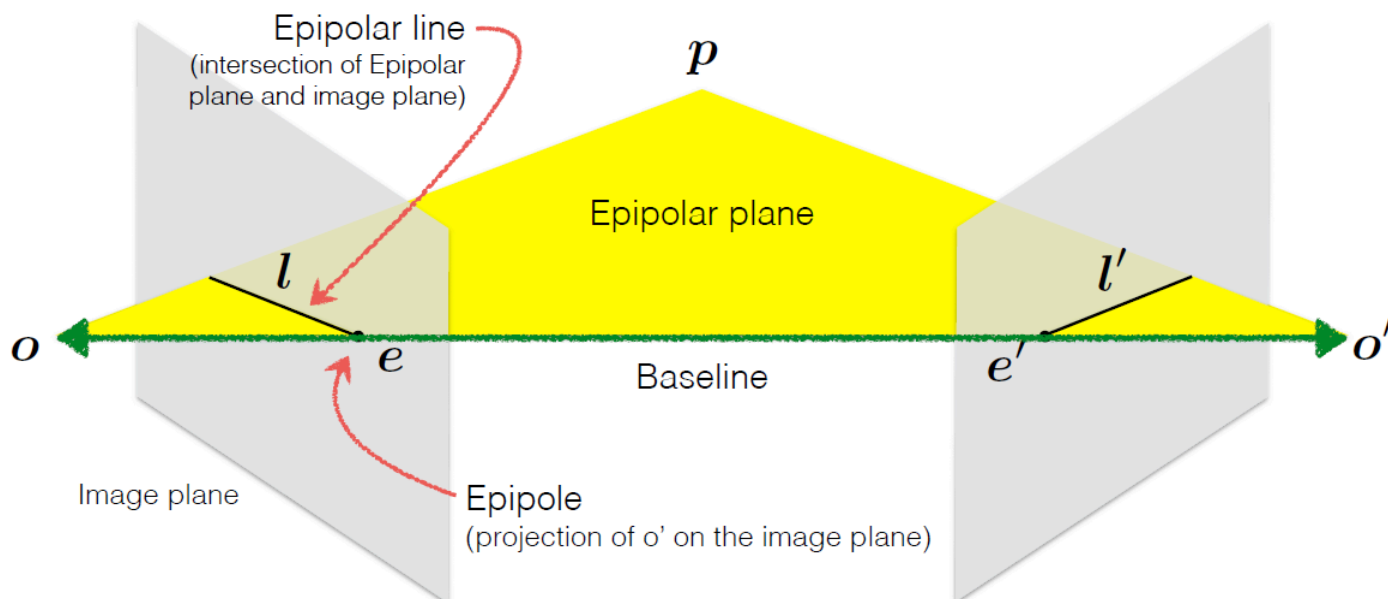


对极几何及单应矩阵

1.本质 矩阵

用两个相机在不同位置拍摄同一物体，两张照片中的景物有重叠部分，那么理论上这两张照片会存在一定的对应关系，本节任务是探索如何描述他们之间的对应关系--对极几何，属于立体视觉的部分。关于相机 成像 模型等部分这里不多介绍。

首先来看几个基本概念



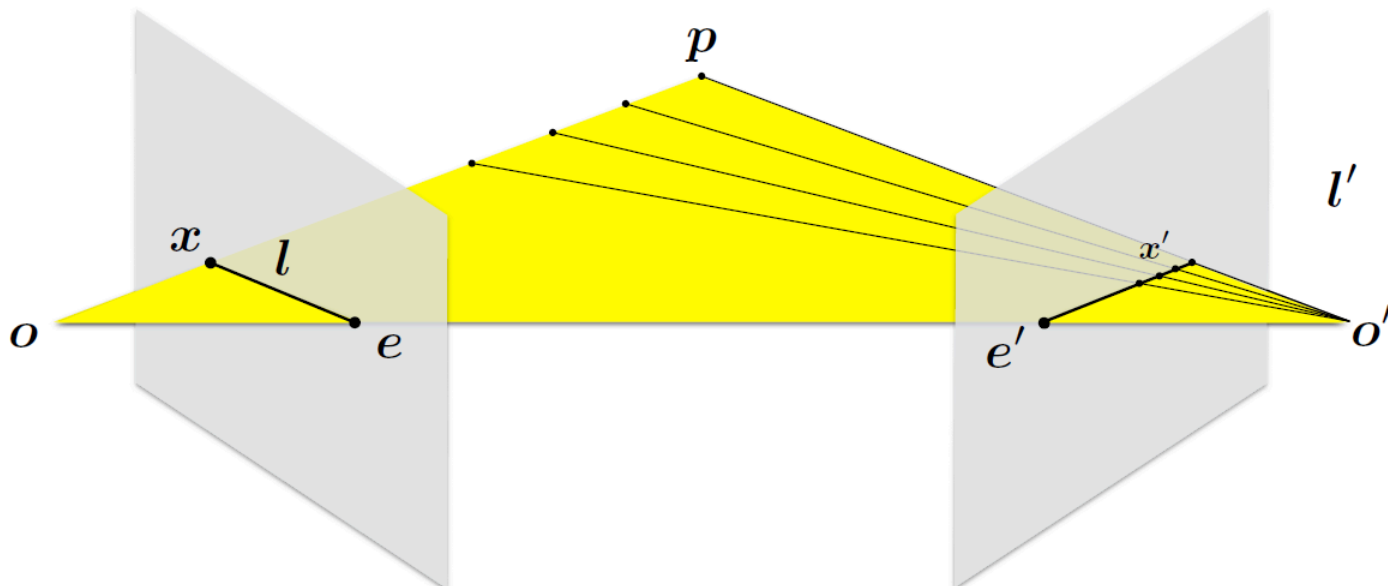
图中的概念有：

极点e：分别是左边相机中心在右图像平面上的像，右相机中心在左像平面上的像。

极平面：两个相机中心和空间中某点p形成的平面。

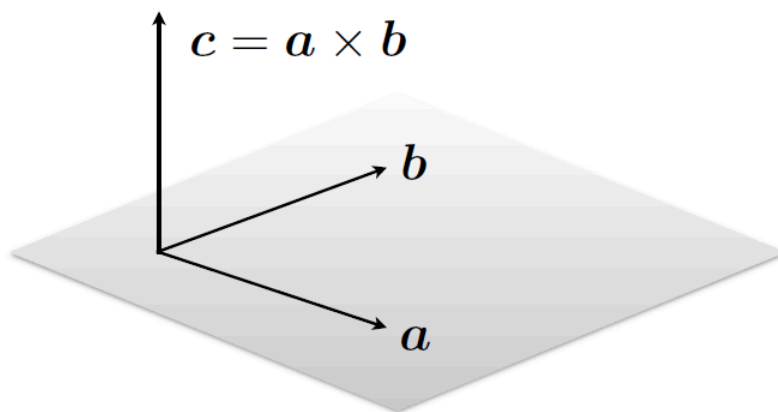
极线l：极平面分别和两个像平面的交线。

对极几何则是描述这几个量之间的对应关系。直观讲，从左图的角度看，如果不知道p点的深度信息，射线op是这个点可能出现的空间位置，因为该射线上的点都会投影到同一个像素点，同时，如果不知道p点的具体位置，那么当在右图的位置看时，极线 l' 就是点p可能出现的位置，即在这条线上的某个地方。如下图所示



Potential matches for x lie on the epipolar line l'

回顾向量的知识

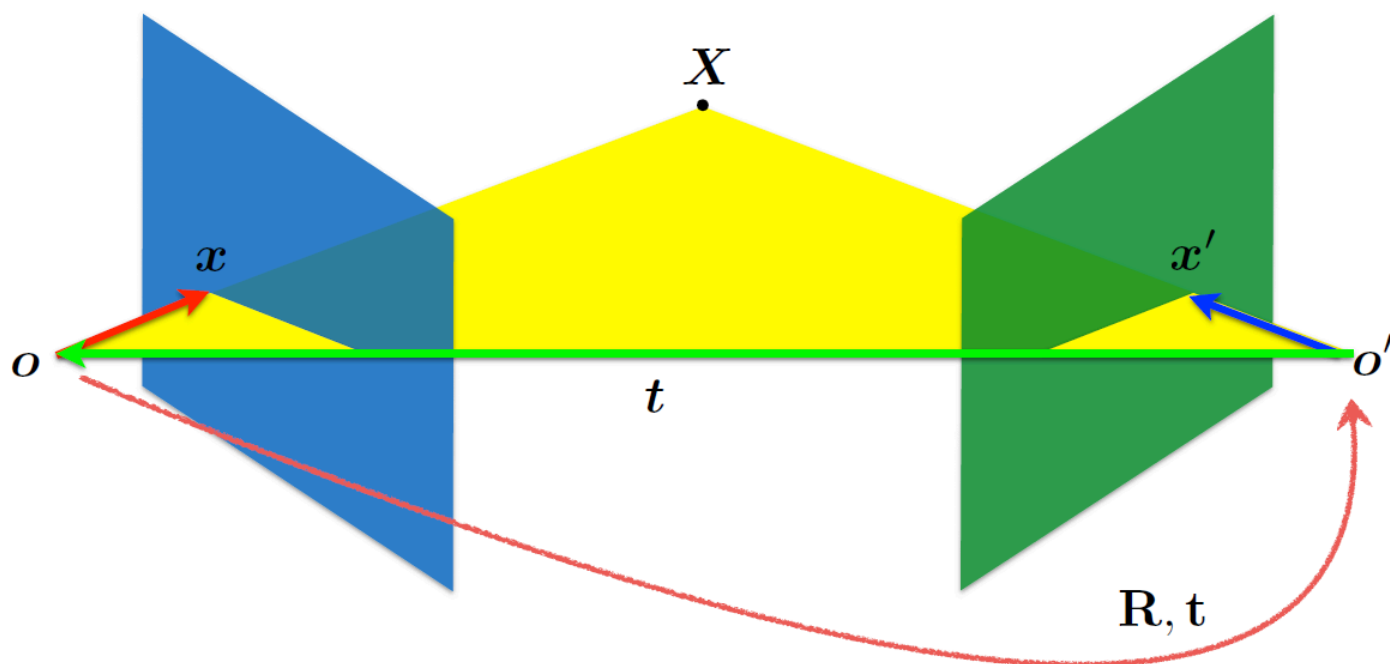


$$c \cdot a = 0$$

$$c \cdot b = 0$$

dot product of two orthogonal vectors is zero

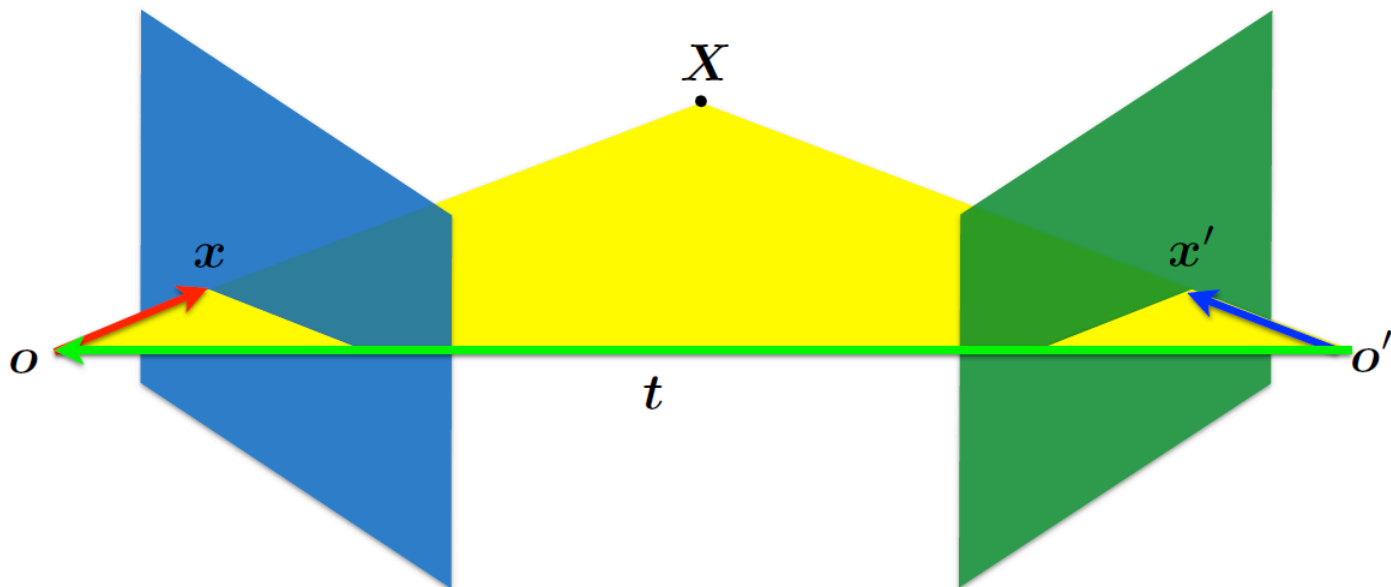
两个正交的向量的内积为0。假设左图到右图的位姿关系由 R 和 t 表示，那么由基本的刚体变换可以得到



$$x' = R(x - t)$$

Camera-camera transform just like **world-camera** transform

另外



If these three vectors are coplanar $\mathbf{x}, \mathbf{t}, \mathbf{x}'$ then

$$\mathbf{x}^\top (\mathbf{t} \times \mathbf{x}) = 0$$

已知三向量 \mathbf{x} , \mathbf{t} 和 \mathbf{x}' 共面，则 \mathbf{t} 和 \mathbf{x} 做外积得到一个垂直该平面的向量，因此上式为零没有问题，接下来

putting it together

rigid motion

$$\mathbf{x}' = \mathbf{R}(\mathbf{x} - \mathbf{t})$$

coplanarity

$$(\mathbf{x} - \mathbf{t})^\top (\mathbf{t} \times \mathbf{x}) = 0$$

$$(\mathbf{x}'^\top \mathbf{R})(\mathbf{t} \times \mathbf{x}) = 0$$

简单的代入即可。然后，向量的外积可以写成矩阵相乘的形式，即将其中一个向量写成反对称矩阵，如下图

Cross product

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

Can also be written as a matrix multiplication

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

Skew symmetric

那么将之前的 \mathbf{t} 和 \mathbf{x} 的外积也写成矩阵的形式可得

putting it together

rigid motion

$$\mathbf{x}' = \mathbf{R}(\mathbf{x} - \mathbf{t})$$

coplanarity

$$(\mathbf{x} - \mathbf{t})^{\top} (\mathbf{t} \times \mathbf{x}) = 0$$

$$(\mathbf{x}'^{\top} \mathbf{R})(\mathbf{t} \times \mathbf{x}) = 0$$

$$(\mathbf{x}'^{\top} \mathbf{R})([\mathbf{t}_{\times}]\mathbf{x}) = 0$$

$$\mathbf{x}'^{\top} (\mathbf{R}[\mathbf{t}_{\times}])\mathbf{x} = 0$$

$$\mathbf{x}'^{\top} \mathbf{E} \mathbf{x} = 0$$

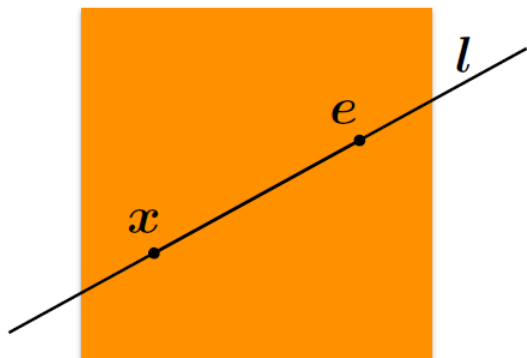
其中的矩阵 \mathbf{E} 就是本质矩阵，最后一个式子为对极约束，形式非常简洁，描述了不同位置拍摄同一个点时这两个点构成的约束，同时可以看到矩阵 \mathbf{E} 由旋转矩阵 \mathbf{R} 和平移向量 \mathbf{t} 组成，对 \mathbf{E} 进行分解即可得到两幅图像的位姿关系，前提是左右图像中的两个点要正确匹配！！

注意：这里的左右两个点 \mathbf{x} 和 \mathbf{x}' 分别为在**相机坐标系下的归一化平面上**的点，即 $\mathbf{x} = [x, y, 1]$ ，而不是像素平面上的点，因此需首先获得空间点在相机系中的坐标值，之后将其归一化到 $z = 1$ 的平面上，才能使用对极约束求解。

前文提到过，对极几何描述的是一个空间点经过对极约束后得到的是其可能出现在第二幅图像中的极线上，这里进行解释：

利用直线的一般方程的知识

$$ax + by + c = 0 \quad \text{in vector form} \quad \mathbf{l} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$



If the point \mathbf{x} is on the epipolar line \mathbf{l} then

$$\mathbf{x}^\top \mathbf{l} = 0$$

当点在直线上时，代入直线方程结果为零。回到对极约束中，这里用 \mathbf{p} 代替上文中左右两图中的 \mathbf{x} 点

- Remember:

$$\mathbf{p}_r^\top \text{Ep}_l = 0$$

$$\tilde{\mathbf{l}}_r = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

\mathbf{p}_r belongs to epipolar line in the right image defined by

$$\tilde{\mathbf{l}}_r = \text{Ep}_l$$

前面说过对极约束中的点是相机系中归一化平面上的点，是三维的点，因此将其和本质矩阵相乘，将得到一个3行1列的向量，而由图可知左图中的点乘以这个向量结果为零，因此可以说点在这个向量构成的直线方程上，同样的

- Remember:

$$p_r^T E p_l = 0$$

$$\tilde{l}_l^T = \begin{bmatrix} a \\ b \\ c \end{bmatrix}^T$$

p_l belongs to epipolar line in the left image defined by

$$\tilde{l}_l = E^T p_r$$

因此，本质矩阵和左像点坐标相乘后得到了一条直线，而该点在右图中的位置就在这条直线上。本质矩阵和右像点相乘后也得到一条直线，该点在左图中的位置也处在这条直线上。

可以看到：本质矩阵将一个点映射到了一条线上，该点在另一幅图像中可能的位置就在这条线上。

本质矩阵的求解

对极约束是一个等式为零的约束，因此对他乘以非零常数后依然成立，这称为本质矩阵的尺度等价性。另一方面，由于平移和旋转各有三个自由度，故 E 共有六个自由度，但由于尺度等价性，因此实际上 E 具有5个自由度。

即最少可用5对点来求解 E ，但是这种解法比较麻烦。考虑 E 的尺度等价性，可以使用8对点来求解，即经典的八点法，八点法利用了 E 的线性性质，因此可以通过线性代数求解。

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

$$x_i x'_i f_{11} + x_i y'_i f_{21} + x_i f_{31} + y_i x'_i f_{12} + y_i y'_i f_{22} + y_i f_{32} + x'_i f_{13} + y'_i f_{23} + f_{33} = 0$$

$$\begin{bmatrix} x'_i & y'_i & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix} = 0$$

Given m point correspondences...

$$\begin{bmatrix} x_1x'_1 & x_1y'_1 & x_1 & y_1x'_1 & y_1y'_1 & y_1 & x'_1 & y'_1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_mx'_m & x_my'_m & x_m & y_mx'_m & y_my'_m & y_m & x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_{11} \\ f_{21} \\ f_{31} \\ f_{12} \\ f_{22} \\ f_{32} \\ f_{13} \\ f_{23} \\ f_{33} \end{bmatrix} = 0$$

Think: how many points do we need?

2.基础矩阵

必须注意本质矩阵描述的是几何意义上的约束，他只是将左右相机内的物理坐标进行了约束。实际处理图像时，最直接得到的是某个点在图像中的像素坐标值，由于像素坐标和归一化坐标通过相机内参矩阵进行联系，因此可得

The essential matrix uses CAMERA coordinates

To use image coordinates we must consider the INTRINSIC camera parameters:

$$\begin{array}{lcl} \text{Pixel coord} & \text{Affine transform matrix} & \text{Camera (film) coord} \\ \text{(row,col)} & \searrow & \swarrow \\ \bar{p}_l = M_l p_l & & p_l = M_l^{-1} \bar{p}_l \\ \bar{p}_r = M_r p_r & & p_r = M_r^{-1} \bar{p}_r \end{array}$$

假设M为相机内参，代入对极约束

$$\begin{aligned}
 p_l &= M_l^{-1} \bar{p}_l \\
 p_r &= M_r^{-1} \bar{p}_r \\
 p_r^T E p_l &= 0 \\
 (M_r^{-1} \bar{p}_r)^T E (M_l^{-1} \bar{p}_l) &= 0 \\
 \bar{p}_r^T (M_r^{-T} E M_l^{-1}) \bar{p}_l &= 0 \\
 \boxed{\bar{p}_r^T F \bar{p}_l = 0}
 \end{aligned}$$

short version: The same equation works in pixel coordinates too!

上图中 F 矩阵即为基础矩阵，他不光包含了旋转和平移的信息，还包含了相机的内参信息，使用的是像素坐标。

3.单应矩阵

单应是射影几何中的概念，又称为射影变换。他把一个射影平面上的点映射到另一个平面对应的位置，并且把直线映射为直线，具有保线性质。与对极几何不同的是，对极几何将点映射到线上，而单应矩阵是点对点的关系。要注意的是单应矩阵的适用场景为：当场景中的特征点都落在同一平面上，比如墙、地面等，此时可用单应性估计运动。其描述方程为

$$x_1 = H x_2$$

其中矩阵 H 为一个3x3的非奇异矩阵，且对矩阵 H 乘以非零的常量依然成立

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \alpha \mathbf{H} \begin{bmatrix} x \\ y \\ w \end{bmatrix}$$

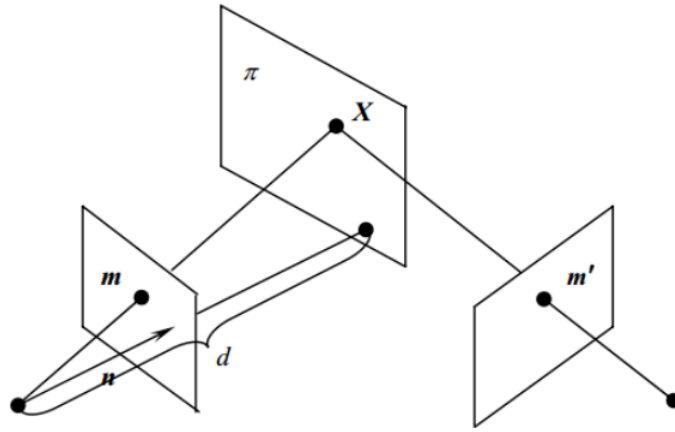
homography $\mathbf{H} = \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix}$
parameters of the transform

假设已经获得了两幅图像之间的单应矩阵，可得

$$\begin{pmatrix} u_1 \\ v_1 \\ 1 \end{pmatrix} = H \begin{pmatrix} u_2 \\ v_2 \\ 1 \end{pmatrix}$$

其中左边为图像1中的点，右边为图像2中的点。其具有很多应用，除了估计运动，还能对图像进行校正，比如斜着拍摄一栋楼，通过单应矩阵，可以生成一张看起来是正对着楼拍摄的照片。

接下来看推导和求解过程



由于是假设特征点处于同一平面，因此首先设平面在第一个相机系下的法向量为 N ，深度为 d ，则平面方程为

$$N^T X_1 = d$$

即

$$\frac{1}{d} N^T X_1 = 1, \forall X_1 \in \pi$$

X_1 ， X_2 分别为点在两个相机系下的坐标，因此

$$X_2 = R X_1 + T$$

将上面两式子结合可得

$$X_2 = R X_1 + T \frac{1}{d} N^T X_1 = (R + T \frac{1}{d} N^T) X_1 = H' X_1$$

因此单应矩阵为

$$H' = R + T \frac{1}{d} N^T$$

可以看到其同样含有旋转和平移信息。

单应矩阵的求解

Matrix Form:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Equations:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

There are 9 numbers h_{11}, \dots, h_{33} , so are there 9 DOF?

No. Note that we can multiply all h_{ij} by nonzero k without changing the equations:

$$\begin{array}{ccc} x' = \frac{kh_{11}x + kh_{12}y + kh_{13}}{kh_{31}x + kh_{32}y + kh_{33}} & \xrightarrow{\text{blue arrow}} & x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} \\ y' = \frac{kh_{21}x + kh_{22}y + kh_{23}}{kh_{31}x + kh_{32}y + kh_{33}} & & y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \end{array}$$

One approach: Set $h_{33} = 1$.

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + \textcircled{1}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + \textcircled{1}}$$

Second approach: Impose unit vector constraint

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Subject to the constraint: $h_{11}^2 + h_{12}^2 + h_{13}^2 + h_{21}^2 + h_{22}^2 + h_{23}^2 + h_{31}^2 + h_{32}^2 + h_{33}^2 = 1$

Setting $h_{33} = 1$

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}$$

Multiplying through by denominator

$$(h_{31}x + h_{32}y + 1)x' = h_{11}x + h_{12}y + h_{13}$$

$$(h_{31}x + h_{32}y + 1)y' = h_{21}x + h_{22}y + h_{23}$$

Rearrange

$$h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' = x'$$

$$h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' = y'$$

$2N \times 8$

8×1

$2N \times 1$

Point 1

x_1

y_1

1

0

0

0

$-x_1x'_1$

$-y_1x'_1$

Point 2

x_2

y_2

1

0

0

0

$-x_2x'_2$

$-y_2x'_2$

Point 3

x_3

y_3

1

0

0

0

$-x_3x'_3$

$-y_3x'_3$

Point 4

x_4

y_4

1

0

0

0

$-x_4x'_4$

$-y_4x'_4$

0

0

0

0

x_4

y_4

1

$-x_4y'_4$

$-y_4y'_4$

h_{11}

h_{12}

h_{13}

h_{21}

h_{22}

h_{23}

h_{31}

h_{32}

x'_1

y'_1

x'_2

y'_2

x'_3

y'_3

x'_4

y'_4

additional points

\bullet

\bullet

\bullet

\bullet

\bullet

\bullet

\bullet

注意：以上做法只有当 h_{33} 即最后一个元素不为零是才能得到正确的解！

以上可以看出，一对匹配点可以得到两项约束，于是自由度为8的单应矩阵可以通过4对匹配点进行求解

对极约束和单应性

- 两幅图像之间的对极约束和场景的结构无关，即对于任意场景结构的图像都是成立的，他不能给出两幅图像上的点的一一对应关系，只能将点映射到线。而单应矩阵可以将点对应到点。
- 单应矩阵，不像对极约束那样完全不限制场景的结构，单应矩阵要求场景中的点都在同一平面上。
- 当相机只有旋转没有平移时，可使用单应矩阵估计运动，因为此时平移为0，计算得到的本质矩阵也为0，进而旋转也为0，得到了错误的解，而使用单应性依然能够正确计算。

<https://blog.csdn.net/u012936940/article/details/80723609>

12/12