[多线程]

thread H
thread F

类成员 → this → ref 引用

两个线程同时计算初始化的两帧间单应矩阵(Homograph Matrix)和基础矩阵(Fundamental Matrix)

Reconstruct H or Reconstruct F

第3周：ORB_SLAM2 课程课件

本课件是公众号 计算机视觉life 旗下课程[《全网最详细的ORB-SLAM2精讲：原理推导+逐行代码分析》](#)（点击可跳转课程详情）的课程课件。谢谢各位学员的支持！

本课程对应的注释代码：https://github.com/electech6/ORBSLAM2_detailed_comments

由于源码注释和课件在持续更新，所以：

如视频课程中注释与上述GitHub中有不同，**以GitHub上最新源码为准。**

如视频课程中课件与本课件不同，**以本课件为准。**

本周课程重点：

1. 掌握基础矩阵F，单应矩阵H的原理（重要），可以手推过程。
2. 理解卡方检验的原理及应用。
3. 掌握检查位姿有效性的方法（重要）。

1. 求解单应矩阵H

1.1 归一化

对应函数Initializer::Normalize

原理参考

Multiple view geometry in computer vision P109 算法4.4

Data normalization is an essential step in the DLT algorithm. It must not be considered optional.

Data normalization becomes even more important for less well conditioned problems,

such as the DLT computation of the fundamental matrix or the trifocal tensor, which

will be considered in later chapters

4.4 Transformation invariance and normalization 1

Objective

Given $n \geq 4$ 2D to 2D point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the 2D homography matrix H such that $\mathbf{x}'_i = \mathtt{H}\mathbf{x}_i$.

Algorithm

(i) **Normalization of x:** Compute a similarity transformation T, consisting of a translation and scaling, that takes points $\mathbf{x}_i$ to a new set of points $\tilde{\mathbf{x}}_i$ such that the centroid of the points $\tilde{\mathbf{x}}_i$ is the coordinate origin $(0,0)^\mathsf{T}$, and their average distance from the origin is $\sqrt{2}$.

(ii) **Normalization of x′:** Compute a similar transformation T′ for the points in the second image, transforming points $\mathbf{x}'_i$ to $\tilde{\mathbf{x}}'_i$.

(iii) **DLT:** Apply algorithm 4.1(*p*91) to the correspondences $\tilde{\mathbf{x}}_i \leftrightarrow \tilde{\mathbf{x}}'_i$ to obtain a homography $\tilde{\mathtt{H}}$.

(iv) **Denormalization:** Set $\mathtt{H} = \mathtt{T}'^{-1}\tilde{\mathtt{H}}\mathtt{T}$.

Algorithm 4.2. *The normalized DLT for 2D homographies.*

为什么要归一化?

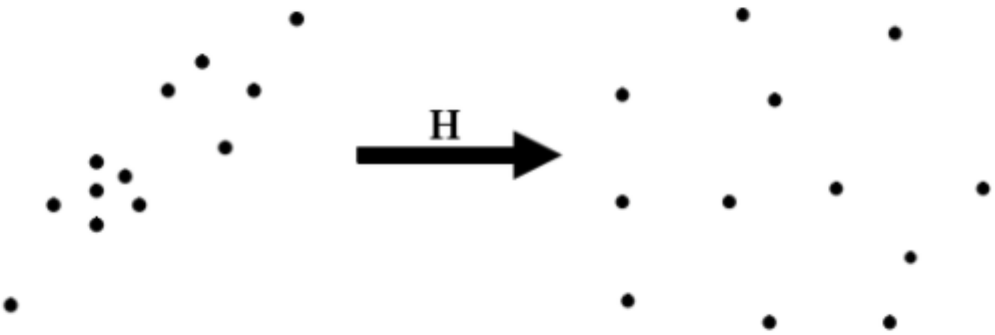
Ah=0

矩阵A是利用8点法求基础矩阵的关键，所以Hartey就认为，利用8点法求基础矩阵不稳定的一个主要原因就是**原始的图像像点坐标组成的系数矩阵A不好造成的**，而造成A不好的原因是**像点的齐次坐标各个分量的数量级相差太大**。基于这个原因，Hartey提出一种改进的8点法，**在应用8点法求基础矩阵之前，先对像点坐标进行归一化处理，即对原始的图像坐标做同向性变换**，这样就可以减少噪声的干扰，大大的提高8点法的精度。

预先对图像坐标进行归一化有以下好处：

- 能够提高运算结果的精度
- 利用归一化处理后的图像坐标，对任何尺度缩放和原点的选择是不变的。归一化步骤预先为图像坐标选择了一个标准的坐标系中，消除了坐标变换对结果的影响。

归一化操作分两步进行，首先对每幅图像中的坐标进行平移（每幅图像的平移不同）使图像中匹配的点组成的点集的质心（Centroid）移动到原点；接着对坐标系进行缩放使得各个分量总体上有一样的平均值，各个坐标轴的缩放相同的



使用归一化的坐标虽然能够在一定程度上消除噪声、错误匹配带来的影响，但还是不够的。

[参考资料]

缩放尺度是为了让噪声对于图像的影像在一个数量级上。

具体归一化操作

一阶矩就是随机变量的期望，二阶矩就是随机变量平方的期望；一阶绝对矩定义为变量与均值绝对值的平均。

向量

$$u_1, u_2, \dots u_N$$

归一化后 ⟶ 放到同一坐标系

x' y' 均值为0，一阶矩为1

期望

$$\bar{u} = E(u)$$

一阶矩绝对矩

$$|\bar{u}| = \sum_{i=0}^{N} |x_i - \bar{u}| / N$$

令新的N维向量维

$$u' = \frac{u - \bar{u}}{|\bar{u}|}$$

疑问：变换矩阵T为何这样？

答案：就是把上述变换用矩阵表示了而已

$\frac{1}{|\bar{u}_x|}$ $\quad$ $\bar{u}_x$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} sX & 0 & -\text{mean}\, X * sX \\ 0 & sY & -\text{mean}\, Y * sY \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$\frac{x}{|\bar{u}_x|} - \frac{\bar{u}_x}{|\bar{u}_x|}$

$\frac{y}{|\bar{u}_y|} - \frac{\bar{u}_y}{|\bar{u}_y|}$

P理解为原点到平面任意一点的向量
也可以理解为平面不过原点，n^TP 即法线对向量的投影得0，用d修正

$n^TP=0$ 成立的话平面要过原点
一般化的话用d替换。

1.2 H矩阵求解原理

单应矩阵

特征点对p_1, p_2，用单应矩阵H_{21} 来描述特征点对之间的变换关系

$$p_2 = H_{21} * p_1$$

$\boxed{n^TP + d = 0}$ 平面约束

$-\frac{n^TP}{d} = 1$

$p_2 = K(RP + t)$

$p_2 = K(RP - \frac{tn^TP}{d})$

$= K(R - \frac{tn^T}{d})P$

$= K(R - \frac{tn^T}{d})K^{-1}p_1$

P能提出来是因为任何一个平面上的点都满足平面方程

我们写成矩阵形式：

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

$Ax=0$

为了化为齐次方程，左右两边同时叉乘p_2，得到

叉积得到与a,b相垂直的向量

$$\begin{vmatrix} i & j & k \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \quad \begin{bmatrix} a_2b_3-a_3b_2 \\ a_3b_1-a_1b_3 \\ a_1b_2-a_2b_1 \end{bmatrix} \begin{matrix} i \\ j \\ k \end{matrix}$$

向量的模为a,b组成的四边形的面积

$$p_2 \times H_{21} * p_1 = 0$$

写成矩阵形式:

$$\begin{bmatrix} 0 & -1 & v_2 \\ 1 & 0 & -u_2 \\ -v_2 & u_2 & 0 \end{bmatrix} \begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

展开计算得到

$$\begin{bmatrix} 0 & -1 & v_2 \\ & & \\ & & \end{bmatrix} \begin{bmatrix} h_1u_1+h_2v_1+h_3 \\ h_4u_1+h_5v_1+h_6 \\ h_7u_1+h_8v_1+h_9 \end{bmatrix} = 0$$

$$v_2 = (h_4 * u_1 + h_5 * v_1 + h_6)/(h_7 * u_1 + h_8 * v_1 + h_9)$$
$$u_2 = (h_1 * u_1 + h_2 * v_1 + h_3)/(h_7 * u_1 + h_8 * v_1 + h_9)$$

写成齐次方程

$$-(h_4 * u_1 + h_5 * v_1 + h_6) + (h_7 * u_1 * v_2 + h_8 * v_1 * v_2 + h_9 * v_2) = 0$$
$$h_1 * u_1 + h_2 * v_1 + h_3 - (h_7 * u_1 * u_2 + h_8 * v_1 * u_2 + h_9 * u_2) = 0$$

一对点两个方程

转化为矩阵形式

$$\begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & u_1 * v_2 & v_1 * v_2 & v_2 \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1 * u_2 & -v_1 * u_2 & -u_2 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = 0$$

等式左边两项分别用A, X表示，则有

求最小二乘解: $\frac{1}{2}(AX)^T(AX) = \frac{1}{2}X^TA^TAX$

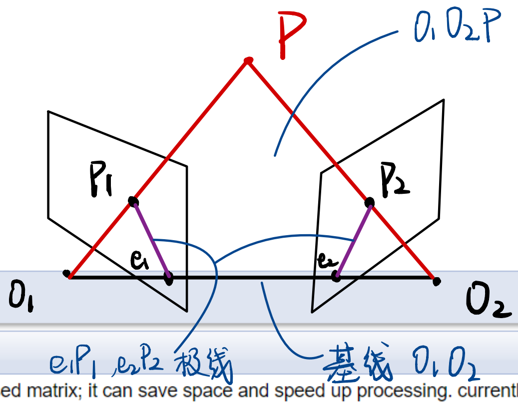
SVD → UWV^T

↳ 8点法 16个方程

$$AX = 0$$

$A \in \mathbb{R}^{16\times 9}$　$U \in \mathbb{R}^{16\times 16}$　$W \in \mathbb{R}^{16\times 9}$　$V \in \mathbb{R}^{9\times 9}$

一对点提供两个约束等式，单应矩阵H总共有9个元素，8个自由度（尺度等价性），所以需要4对点提供8个约束方程就可以求解。



e_1, e_2为基线与成像平面交点

O_1, O_2是光心

1.3 SVD分解

§ Flags

enum cv::SVD::Flags

Enumerator	
MODIFY_A	allow the algorithm to modify the decomposed matrix; it can save space and speed up processing. currently ignored.
NO_UV	indicates that only a vector of singular values `w` is to be processed, while u and vt will be set to empty matrices
FULL_UV	when the matrix is not square, by default the algorithm produces u and vt matrices of sufficiently large size for the further A reconstruction; if, however, FULL_UV flag is specified, u and vt will be full-size square orthogonal matrices.

1.4 为什么V^T的第9个奇异向量就是最优解?

Ah=0 对应的代价函数

$$f(h) = \frac{1}{2}(Ah)^T(Ah) = \frac{1}{2}h^TA^TAh$$

h是未知数

对h求导

最优解是导数为0

$$\frac{df}{dh} = 0$$
$$A^TAh = 0$$

方阵

会遍历每对匹配点，计算得分（重投影），取得分最高的点对对应的H矩阵

问题就转换为求A^TA的最小特征值向量

正交矩阵　特征值

$$A^TA = ((UDV^T)^T(UDV^T)) = VD^TU^TUDV^T = VD^TDV^T$$

可见A^TA的特征向量就是V^T的特征向量。因此求解得到V 之后取出最后一行奇异值向量作为f的最优值，然后整理成3维矩阵形式。（其实其他行的奇异值向量也是一个解，但是不是最优解）

$Ax=0$ 近似解 $x^+ = \arg\min \|Ax\|^2$, subjeoe to $\|x\|^2=1$

若x是Ax=0的一个解 故kx也是，$\|x\|^2=1$

$L(x,\lambda) = \|Ax\|^2 + \lambda(1-\|x\|^2) = x^TA^TAx + \lambda(1-x^Tx)$

$\rightarrow \frac{dL(x,\lambda)}{dx} = 2A^TAx - 2\lambda x = 0$

$\rightarrow A^TAx = \lambda x$

$\rightarrow \underbrace{\|Ax\|^2}_{\arg\min} = x^TA^TAx = x^T\lambda x = \lambda x^Tx = \lambda$

所以，对于超定方程 Ax=0 的解是 A^TA 最小特征值对应的特征向量。

（SVD分解）

$A = UDV^T$

正交矩阵有保范性
即正交变换不改变二范数

$\rightarrow |Ax| = |UDV^Tx| = |DV^Tx|$

$\rightarrow |V^Tx| = |x| = 1$ 令 $y = V^Tx$

$\rightarrow \arg\min|DV^Tx| = \arg\min|Dy|$

故 D最小特征值（降序对角矩阵）

$D = \begin{vmatrix} \ddots & \\ & 0 \end{vmatrix}$　$y = \begin{vmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{vmatrix}$

能使|Dy|最小

Ax表示A的列向量空间，是A的列向量的线性组合

Ax=0 不成立，令Ax=b 是在$Ax^*=0$的张量空间外，Ax^*是Ax的投影

$b-Ax^*$ 一定与张量空间中任一向量垂直

Ax^*所在的张量空间S的A列向量的线性组合

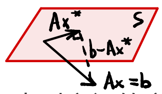
故 $A^T(b-Ax^*)=0$

$A^Tb = A^TAx^*$

$x^* = (A^TA)^{-1}A^Tb$

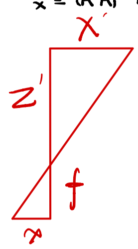
2. 求解基础矩阵F

2.1 推导F矩阵约束方程



特征点对p_1, p_2，用基础矩阵F_{21} 来描述特征点对之间的变换关系

$$p_2^T * F_{21} * p_1 = 0$$



$x = f_x\frac{X'}{Z'} + c_x$

$y = f_y\frac{Y'}{Z'} + c_y$

我们写成矩阵形式：

$$\begin{bmatrix} u_2 & v_2 & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0$$

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

$P_1 = \begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \end{bmatrix}$

为方便计算先展开前两项得到

$$a = f_1 * u_2 + f4 * v_2 + f_7;$$
$$b = f_2 * u_2 + f_5 * v_2 + f_8;$$
$$c = f_3 * u_2 + f_6 * v_2 + f_9;$$

对极约束：

① $Z_1p_1 = KP_1$　$Z_2p_2 = K(RP_1+t)$

$p_1 \simeq KP_1$　$p_2 \simeq K(RP_1+t)$

② $x_1 = K^{-1}p_1$　$x_2 = K^{-1}p_2$

$= Rx_1 + t$

$\frac{P_1}{Z_1}$ 归一化平面投影

③ $t^\wedge x_2 = t^\wedge Rx_1$

④ $x_2^Tt^\wedge x_2 = x_2^Tt^\wedge Rx_1$

$0 =$

那么，上面的矩阵可以化为

$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} u_1 \\ v_1 \\ 1 \end{bmatrix} = 0$$

⑤ $p_2^TK^{-T}t^\wedge RK^{-1}p_1 = 0$

基础矩阵F

本质矩阵

展开后：

$$a * u_1 + b * v_1 + c = 0$$

带入前面a,b,c表达式子整理得到

$$f_1 * u_1 * u_2 + f_2 * v_1 * u_2 + f_3 * u_2 + f_4 * u_1 * v_2 + f_5 * v_1 * v_2 + f_6 * v_2 + f_7 * u_1 + f_8 * v_1 + f_9 = 0$$

转化为矩阵形式

$$
\begin{bmatrix} u_1*u_2 & v_1*u_2 & u_2 & u_1*v_2 & v_1*v_2 & v_2 & u_1 & v_1 & 1 \end{bmatrix}
\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = 0
$$

等式左边两项分别用A, f表示，则有

$$
Af = 0
$$

一对点提供一个约束方程，基础矩阵F总共有9个元素，7个自由度（尺度等价性，秩为2），所以8对点提供8个约束方程就可以求解F。

2.2 SVD

9阶矩阵，秩为2，那基础解系线性无关的向量个数是 9-2=7个

→ 秩为2 $\begin{bmatrix} \text{——} & & & & \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \; \end{bmatrix} = 0$

SVD分解结果

$$
A = UDV^T
$$

假设我们使用8对点求解，A 是 8x9 矩阵，分解后

U 是左奇异向量，它是一个8x8的 正交矩阵，

V 是右奇异向量，是一个 9x9 的正交矩阵，V^T是V的转置

D是一个8 x 9 对角矩阵，除了对角线其他元素均为0，对角线元素称为奇异值，一般来说奇异值是按照从大到小的顺序降序排列。因为每个奇异值都是一个残差项，因此最后一个奇异值最小，其含义就是最优的残差。因此其对应的奇异值向量就是最优值，即最优解。

V^T中的每个列向量对应着D中的每个奇异值，最小二乘最优解就是V^T对应的第9个列向量，也就是基础矩阵F的元素。这里我们先记做 Fpre，因为这个还不是最终的F

2.3 F矩阵秩为2

$F = K^{-T} t^\wedge R K^{-1}$

K秩为3　R秩为3　$t^\wedge$秩为2　→ 故F秩为2

基础矩阵 F 有个很重要的性质，就是秩为2，可以进一步约束求解准确的F

上面的方法使用V^T对应的第9个列向量构造的Fpre 秩通常不为2，我们可以继续进行SVD分解。

$$
F_{pre} = UDV^T = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} V^T
$$

$$
t^\wedge = \begin{vmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & a_1 \\ -a_2 & -a_1 & 0 \end{vmatrix}
\rightarrow \begin{vmatrix} 0 & -a_3 & a_2 \\ a_2a_3 & 0 & a_1a_2 \\ -a_2a_3 & -a_1a_3 & 0 \end{vmatrix}
\rightarrow \begin{vmatrix} 0 & -a_3 & a_2 \\ a_2a_3 & 0 & a_1a_2 \\ 0 & -a_1a_3 & a_1a_2 \end{vmatrix}
\rightarrow \begin{vmatrix} 0 & -a_1a_3 & a_1a_2 \\ a_2a_3 & 0 & a_1a_2 \\ 0 & 0 & 0 \end{vmatrix}
$$

其最小奇异值人为置为0，这样F矩阵秩为2

$$
F = UDV^T = U \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T
$$

此时的F就是最终得到的基础矩阵。

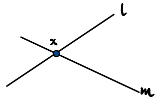
点在线上用齐次坐标点乘表示：

直线 l: $ax+by+c=0$，用向量表示 $l=(a,b,c)^T$，

那么 $ax+by+c*1=0 \Rightarrow (a,b,c)^T(x,y,1) = l^T * p' = 0$

直线的参数向量与平面内一点的齐次坐标点乘为0，即点在该线上。

直线与直线 l, m 的叉乘表示交点 x：



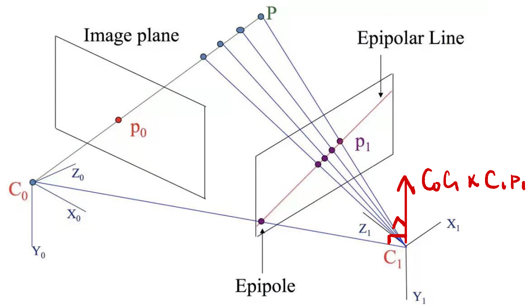
$$l^T * (l \times m) = l^T * x = 0$$

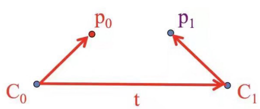
$$m^T * (l \times m) = m^T * x = 0$$

点乘-垂直项为0

垂直于 l, m，大小是两向量构造的四边形的面积

用 geometry 理解 epipolar constraints





$$\overrightarrow{C_0p_0} \cdot \left(\overrightarrow{C_0C_1} \times \overrightarrow{C_1p_1}\right) = 0$$

C_0-C_1-P_0-P_1 在极平面上

故在 $(\overrightarrow{C_1P_1} \times \overrightarrow{C_0C_1}) \cdot \overrightarrow{C_0P_0} = 0$ (1)

$\overrightarrow{C_1P_1}$ 和 $\overrightarrow{C_0P_0}$ 可以表示为在 C_1 坐标系和 C_0 坐标系下的 $\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix}$ 和 $\begin{pmatrix} x_0 \\ y_0 \\ 1 \end{pmatrix}$ 的方向向量，把两个成像平面放在归一化平面。

与起始点无关，故不考虑移动（在两个坐标系变换时）

$\overrightarrow{C_0P_0}$ 和 $\overrightarrow{C_1p_1}$ 方向只要不变，(1)式仍满足

$\overrightarrow{C_0C_1}$ 是光心 C_0 到 C_1 的平移，用 t 表示，现在把 $\overrightarrow{C_1p_1}$ 也移到 C_0 坐标系下

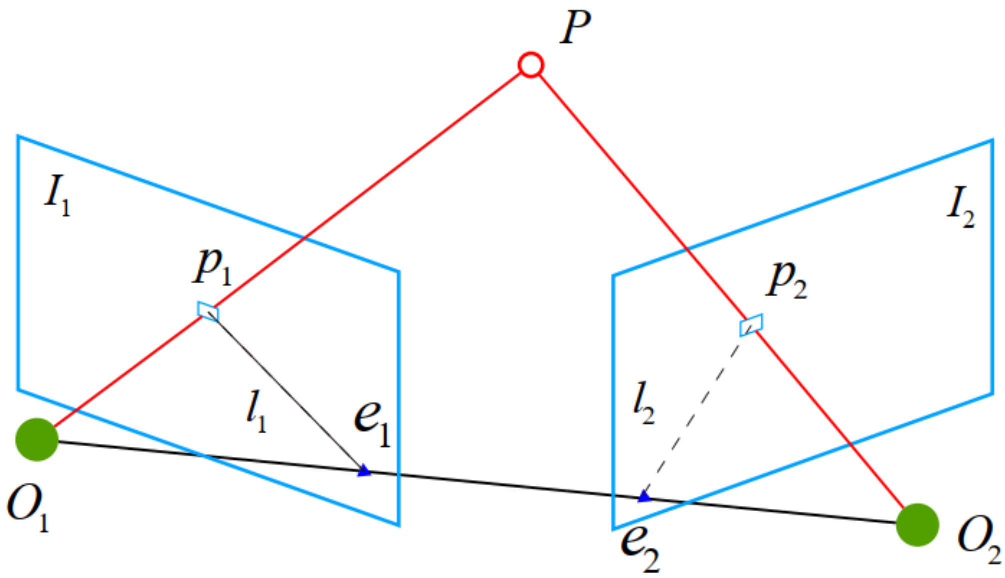
有：$p_0 \cdot (\underline{t \times R}_{12} p_1) = 0$

本质矩阵 E

p_0 在极线上，点与直线参数向量点乘为0，说明点在直线上，故 Ep_1 就是极线方程。

2.4 对极约束

对极约束（极线约束）示意图



3 卡方检验

3.1 为什么要引用卡方检验?

以特定概率分布为某种情况建模时，事物长期结果较为稳定，能够清晰进行把握。比如抛硬币实验。

但是期望与事实存在差异怎么办？偏差是正常的小幅度波动？还是建模错误？此时，利用卡方分布**分析结果**，排除**可疑结果**。

简单来说：当事实与期望不符合情况下使用卡方分布进行检验，看是否系统出了问题，还是属于正常波动

3.2 卡方分布用途?

检查实际结果与期望结果之间何时存在显著差异。
1、检验拟合程度：也就是说可以检验一组给定数据与指定分布的吻合程度。如：用它检验抽奖机收益的观察频数与我们所期望的吻合程度。

2、检验两个变量的独立性：通过这个方法检查变量之间是否存在某种关系。

3.3 卡方分布假设检验步骤?

1、确定要进行检验的假设（H0）及其备择假设H1.
2、求出期望E.
3、确定用于做决策的拒绝域（右尾）.
4、根据**自由度**和**显著性水平**查询检验统计量临界值.
5、查看检验统计量是否在拒绝域内.
6、做出决策.

3.4 一个例子：抽奖机之谜

抽奖机，肯定都不陌生，现在一些商场超市门口都有放置。正常情况下出奖概率是一定的，综合来看，商家收益肯定大于支出。

倘若突然某段时间内总是出奖，甚是反常，那么到底是某阶段是小概率事件还是有人进行操作了？**抽奖机怎么了？**针对这种现象或者类似这种现象问题则可以借助卡方进行检验，暂且不着急如何检验，还是补充一下基础知识，再逐步深入解决问题。【常规事件中出现非常规现象，如何检查问题所在的情况下使用卡方分布】

下面是某台抽奖机的期望分布，其中X代表每局游戏的净收益（每局独立事件）：

每局2美元，如果什么也赢不到的话，你就损失2美元。

如果中了头奖，净收益就是98美元。

x	-2	23	48	73	98
P(X = x)	0.977	0.008	0.008	0.006	0.001

实际观察中玩家收益的频数为：

频数指出每种收益的发生次数。

x	-2	23	48	73	98
频数	965	10	9	9	7

目的：在5%的显著性水平下，看看能否有足够证据证明判定抽奖机被人动了手脚。

要检验的原假设是什么？备择假设是什么？

H_0：老虎机每局收益符合如下概率分布。

x	-2	23	48	73	98
P(X = x)	0.977	0.008	0.008	0.006	0.001

H_1：老虎机每局收益不符合以上概率分布。

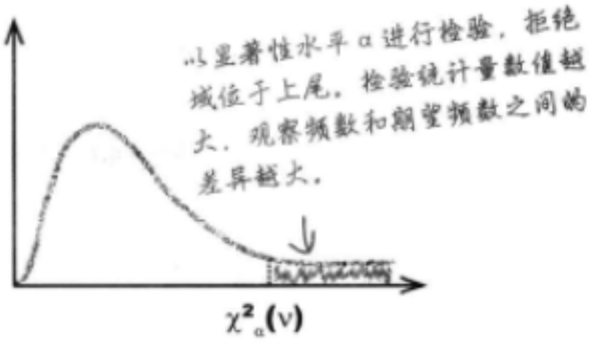
3.5 什么是显著性水平?

显著性水平是估计总体参数落在某一区间内，可能犯错误的概率，用α表示。

显著性水平是假设检验中的一个概念，是指当原假设为正确时人们却把它拒绝了的概率或风险。它是公认的小概率事件的概率值，必须在每一次统计检验之前确定，通常取**α=0.05或α=0.01**。这表明，当作出接受原假设的决定时，其**正确的可能性（概率）为95%或99%**。

卡方分布指出观察频数与期望频数之间差异显著性，和其他假设一样，这取决于显著性水平。

1、显性水平α进行检验，则写作：（常用的显著性水平1%和5%）
2、检测标准：卡方分布检验是单尾检验且是右尾，右尾被作为**拒绝域**。于是通过查看检验统计量是否位于右尾的拒绝域以内，来判定期望分布得出结果的可能性。



3、卡方概率表的使用：卡方临界值表是给定可以查询的

问题简述

抽奖机平常收益者总是商家，突然一段时间总是出奖。本来小概率事件频发，我们利用卡方的检验拟合优度看看能否有足够证据证明判定抽奖机被人动了手脚

3.6 卡方分布是什么?

通过一个检验统计量来比较**期望结果**和**实际结果**之间的差别，然后得出观察频数极值的发生概率。
计算统计量步骤： （期望频数总和与观察频数总和相等）
1、表里填写相应的观察频数和期望频数
2、利用卡方公式计算检验统计量：

$$\chi^2 = \Sigma\frac{(O-E)^2}{E}$$

O 代表观察到的频数，也就是实际发生的频数。E代表期望频数。

检验统计量χ^2意义：O与E之间差值越小，检验统计量χ^2越小。以E为除数，令差值与期望频数成比例。

卡方检验的标准： 如果统计量值χ^2很小，说明观察频数和期望频数之间的差别不显著，统计量越大，差别越显著。

期望频数E的计算

期望频数=（观察频数之和（1000））X （每种结果的概率） 如：X=(-2)的期望频数：977=（0.977）X（1000）

算出每个x值的实际频率与根据概率分布得出的期望频率进行比较

x	观察频数	期望频数
-2	965	977
23	10	8
48	9	8
73	9	6
98	7	1

用每种结果的概率乘以总频数1000，可得期望频数。

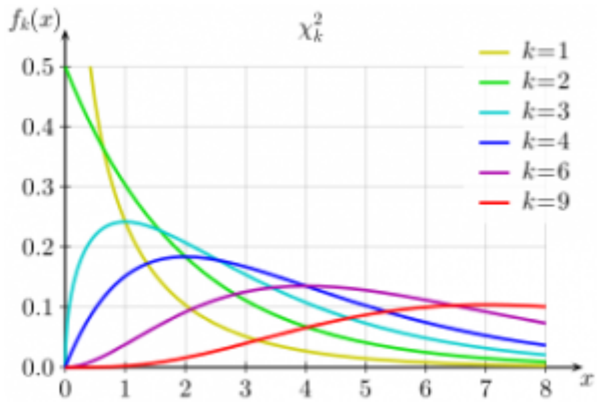
利用抽奖机的观察频率和期望频率表计算检验统计量

$$\chi^2 = (965 - 977)^2/977 + (10 - 8)^2/8 + (9 - 8)^2/8 + (9 - 6)^2/6 + (7 - 1)^2/1$$
$$= (-12)^2/977 + 2^2/8 + 1^2/8 + 3^2/6 + 6^2$$
$$= 144/977 + 4/8 + 1/8 + 9/6 + 36$$
$$= 0.147 + 0.5 + 0.125 + 1.5 + 36$$
$$= 38.272$$

3.7 根据自由度和显著性水平查询检验统计量临界值

自由度的影响

自由度:用于计算检验统计量的独立变量的数目。



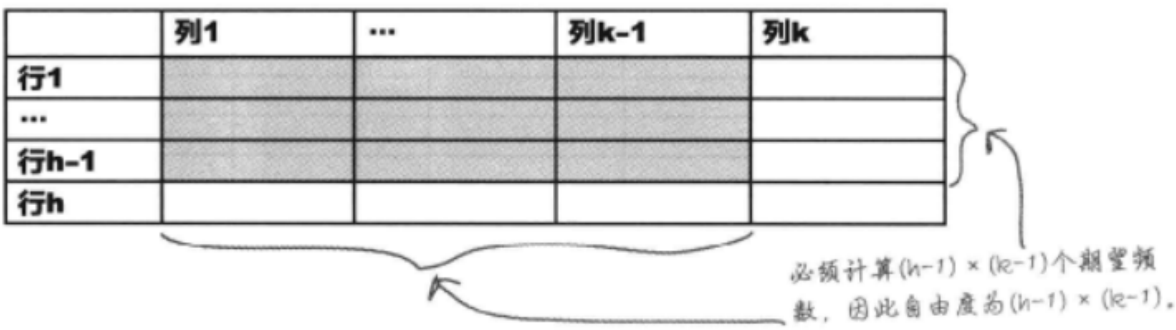
当自由度等于1或者2时：卡方分布先高后低的平滑曲线，检验统计量等于较小值的概率远远大于较大值的概率，即观察频数有可能接近期望频数。

当自由度大于2时：卡方分布先低后高再低，其外形沿着正向扭曲，但当自由度很大时，图形接近正态分布。

自由度的计算，

对于单行或单列：自由度 = 组数-限制数

对于表格类：自由度 = （行数 - 1) * (列数 - 1)



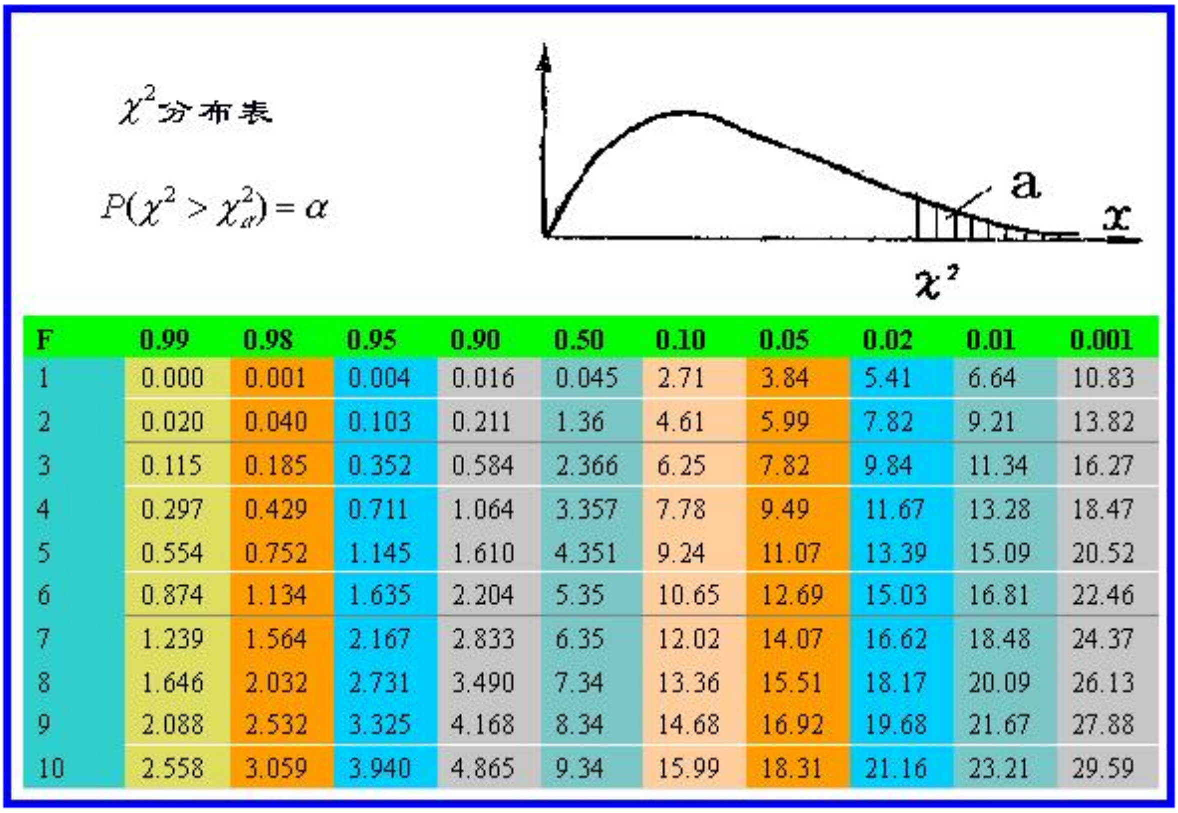
对于 抽奖机的例子，自由度为5-1=4

自由度为4, 5%显著水平的拒绝域是多少?

查表，自由度F=4，显著性为0.05，对应拒绝域 为

$$\chi^2 > 9.49$$

也就是说检验统计量大于9.49 位于拒绝域内



F	0.99	0.98	0.95	0.90	0.50	0.10	0.05	0.02	0.01	0.001
1	0.000	0.001	0.004	0.016	0.045	2.71	3.84	5.41	6.64	10.83
2	0.020	0.040	0.103	0.211	1.36	4.61	5.99	7.82	9.21	13.82
3	0.115	0.185	0.352	0.584	2.366	6.25	7.82	9.84	11.34	16.27
4	0.297	0.429	0.711	1.064	3.357	7.78	9.49	11.67	13.28	18.47
5	0.554	0.752	1.145	1.610	4.351	9.24	11.07	13.39	15.09	20.52
6	0.874	1.134	1.635	2.204	5.35	10.65	12.69	15.03	16.81	22.46
7	1.239	1.564	2.167	2.833	6.35	12.02	14.07	16.62	18.48	24.37
8	1.646	2.032	2.731	3.490	7.34	13.36	15.51	18.17	20.09	26.13
9	2.088	2.532	3.325	4.168	8.34	14.68	16.92	19.68	21.67	27.88
10	2.558	3.059	3.940	4.865	9.34	15.99	18.31	21.16	23.21	29.59

决策原则

如果位于拒绝域内我们拒绝原假设H0，**接受**H1。
如果不在拒绝域内我们**接受**原假设H0，拒绝H1

检验统计量38.272 > 9.49 位于拒绝域内

于是拒绝原假设：抽奖机每局收益如何概率分布，也就是说**抽奖机被人动了手脚**

检验统计量拒绝域内外判定：

1、求出检验统计量a
2、通过自由度和显著性水平查到拒绝域临界值b
3、a>b则位于拒绝域内，反之，位于拒绝域外。

3.8 ORB-SLAM2中的卡方检测剔除外点策略

误差的定义：

就特征点法的视觉SLAM而言，一般会计算重投影误差。具体而言，记 $\mathbf{u}$ 为特征点的2D位置， $\bar{\mathbf{u}}$ 为由地图点投影到图像上的2D位置。重投影误差为

$$\mathbf{e} = \mathbf{u} - \bar{\mathbf{u}} \qquad (1)$$

重投影误差服从高斯分布

$$\mathbf{e} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \qquad (2)$$

其中，协方差 $\boldsymbol{\Sigma}$ 一般根据**特征点提取的金字塔层级**确定。具体的，记提取ORB特征时，图像金字塔的每层缩小尺度为 s (ORB-SLAM中为1.2)。在ORB-SLAM中假设第0层的标准差为 p 个pixel (ORB-SLAM中设为了1个pixel)；那么，一个在金字塔第 n 层提取的特征的重投影误差的协方差为

$$\boldsymbol{\Sigma} = (s^n \times p)^2 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (3)$$

回头再看一下式(1)中的误差是一个2维向量，阈值也不好设置吧。那就把它变成一个标量，计算向量的内积 r （向量元素的平方和）。但是，这又有一个问题，不同金字塔层的特征点都用同一个阈值，是不是不合理呢。于是，在计算内积的时候，利用协方差进行加权（协方差表达了不确定度）。金字塔层级越高，特征点提取精度越低，协方差Σ越大，那么就有了

$$r = \mathbf{e}^T \boldsymbol{\Sigma}^{-1} \mathbf{e} \qquad (4)$$

利用协方差加权，起到了归一化的作用。具体的(4)式，可以变为

$$r = (\boldsymbol{\Sigma}^{-\frac{1}{2}} \mathbf{e})^T (\boldsymbol{\Sigma}^{-\frac{1}{2}} \mathbf{e}) \qquad (5)$$

而

$$(\boldsymbol{\Sigma}^{-\frac{1}{2}} \mathbf{e}) \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$$

H矩阵算的是重投影误差，2维的，有2个自由度。
F矩阵算的是kp到极线的距离，是1维的，1个自由度。

为多维**标准正态分布。**

也就是说**不同金字塔层提取的特征，计算的重投影误差都被归一化了，或者说去量纲化了**，那么，我们只用一个阈值就可以了。

可见：

金字塔层数越高，图像分辨率越低，特征提取的精度也就越低，因此协方差越大

单目投影为2自由度，在0.05的显著性水平（也就是95%的准确率）下，卡方统计量阈值为5.99

双目投影为3自由度，在0.05的显著性水平（也就是95%的准确率）下，卡方统计量阈值为7.81

双目匹配到的特征点在右图中的x坐标为u_{xr}，重投影后计算得到特征点左图的x坐标u_{xl}，根据视差

$$disparity = \frac{baseline * fx}{depth}$$

从而得到重投影后右图中特征点x坐标

$$u_{xr} = u_{xl} - disparity$$

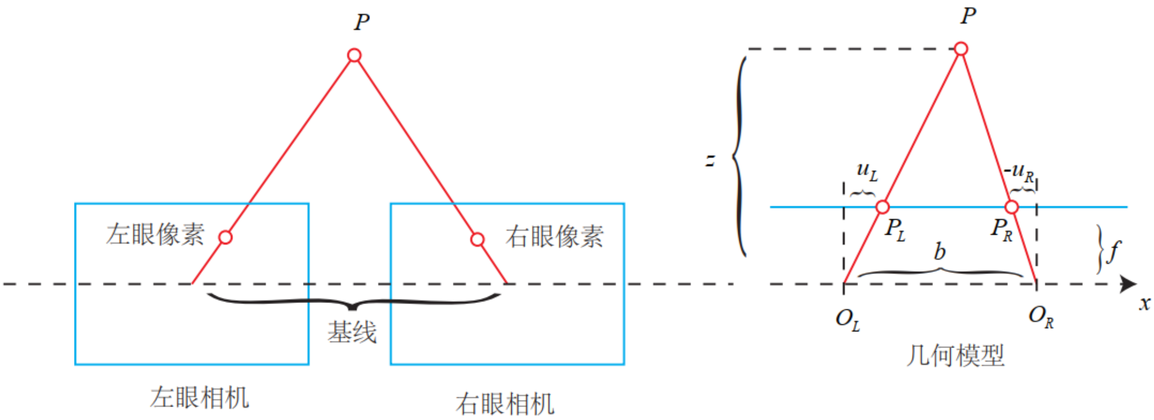
disparity就是另一个自由度

LocalMapping.cc 里面

```
const float &sigmaSquare1 = mpCurrentKeyFrame->mvLevelSigma2[kp1.octave];
            const float x1 = Rcw1.row(0).dot(x3Dt)+tcw1.at<float>(0);
            const float y1 = Rcw1.row(1).dot(x3Dt)+tcw1.at<float>(1);
            const float invz1 = 1.0/z1;
```

```
            if(!bStereo1)
            {
                float u1 = fx1*x1*invz1+cx1;
                float v1 = fy1*y1*invz1+cy1;
                float errX1 = u1 - kp1.pt.x;
                float errY1 = v1 - kp1.pt.y;
                // 基于卡方检验计算出的阈值（假设测量有一个像素的偏差）自由度2
                if((errX1*errX1+errY1*errY1)>5.991*sigmaSquare1)
                    continue;
            }
            else
            {
                float u1 = fx1*x1*invz1+cx1;
                float u1_r = u1 - mpCurrentKeyFrame->mbf*invz1;     // 根据视差公式计算假想的右目坐标
                float v1 = fy1*y1*invz1+cy1;
                float errX1 = u1 - kp1.pt.x;
                float errY1 = v1 - kp1.pt.y;
                float errX1_r = u1_r - kp1_ur;
                // 自由度为3
                if((errX1*errX1+errY1*errY1+errX1_r*errX1_r)>7.8*sigmaSquare1)
                    continue;
            }
```

双目相机



$$\frac{z-f}{z}=\frac{b-u_L+u_R}{b}.$$

$$z=\frac{fb}{d},\quad d=u_L-u_R.$$

[参考1](), [参考2]()

4 检查位姿的有效性

$H = K(R - \frac{nt}{d})K^{-1}$
$A = K^{-1} H K$
H矩阵恢复根据1988年《Motion and Structure from Motion in a Piecewise Planar Environment》中的方法在不讨论A的奇异值 d_1, d_2, d_3 下有8组 R, t, n 的解

针对函数Initializer::CheckRT

第2个相机光心在第1个相机下的坐标：

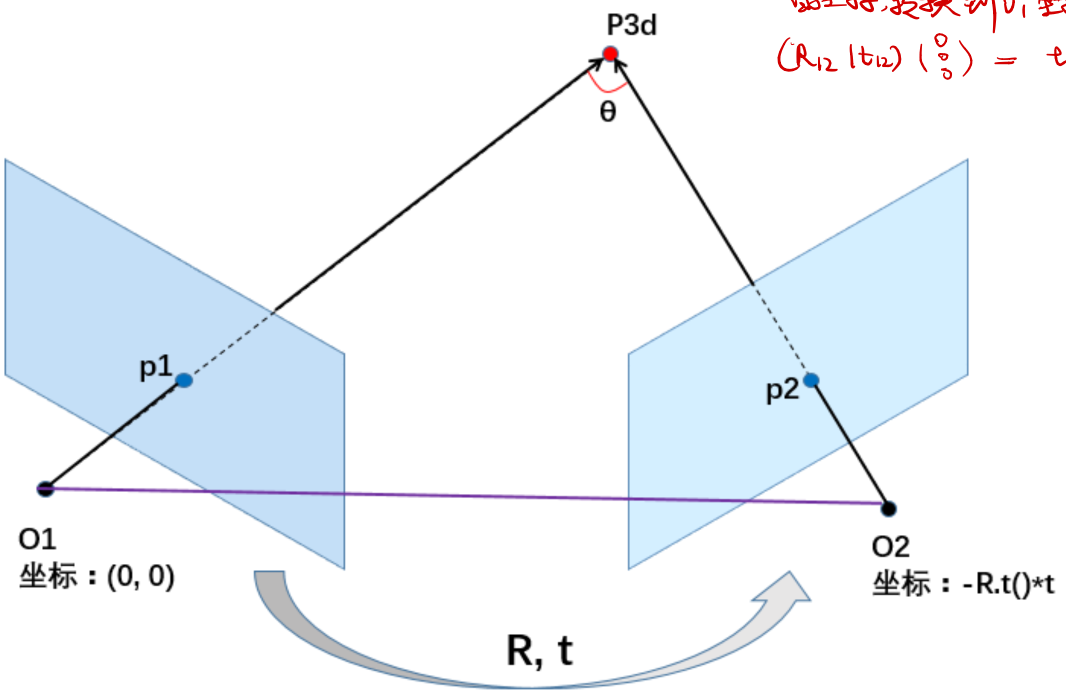
$$T_{21} = \begin{bmatrix} R_{21} & t_{21} \\ 0^T & 1 \end{bmatrix}$$

$$T_{12} = T_{21}^{-1} = \begin{bmatrix} R_{21}^T & -R_{21}^T t_{21} \\ 0^T & 1 \end{bmatrix}$$

$$R_{12} = R_{21}^T$$

$$t_{12} = -R_{21}^T t_{21}$$

(0, 0, 0) 光心 O_2 在 O_2 坐标系下的坐标，转换到 O_1 坐标系
$(R_{12} | t_{12}) \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = t_{12}$



向量余弦夹角：cos(θ) = a•b / |a||b|

5 单目投影恢复3D点

匹配特征点对 x, x'，P, P' 分别是投影矩阵，他们将同一个空间点X投影到图像上的点 x_1, x_2

描述为

$$x = \lambda * P * X$$
$$x' = \lambda * P' * X$$

两个表达式类似，我们以一个通用方程来描述

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

为方便推导，简单记为

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda \begin{bmatrix} - & P_0 & - \\ - & P_1 & - \\ - & P_2 & - \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

为了化为齐次方程，左右两边同时叉乘，得到

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{\times} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{\times} \lambda \begin{bmatrix} - & P_0 & - \\ - & P_1 & - \\ - & P_2 & - \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}_{\times} \lambda \begin{bmatrix} - & P_0 & - \\ - & P_1 & - \\ - & P_2 & - \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & -1 & y \\ 1 & 0 & -x \\ -y & x & 0 \end{bmatrix} \begin{bmatrix} - & P_0 & - \\ - & P_1 & - \\ - & P_2 & - \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} yP_2 - P_1 \\ P_0 - xP_2 \\ xP_1 - yP_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

两对匹配点

$$\begin{bmatrix} yP_2 - P_1 \\ P_0 - xP_2 \\ y'P_2' - P_1' \\ P_0' - x'P_2' \end{bmatrix} X = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

等式左边两项分别用A, X表示，则有

$$AX = 0$$

SVD求解，右奇异矩阵的最后一行就是最终的解
