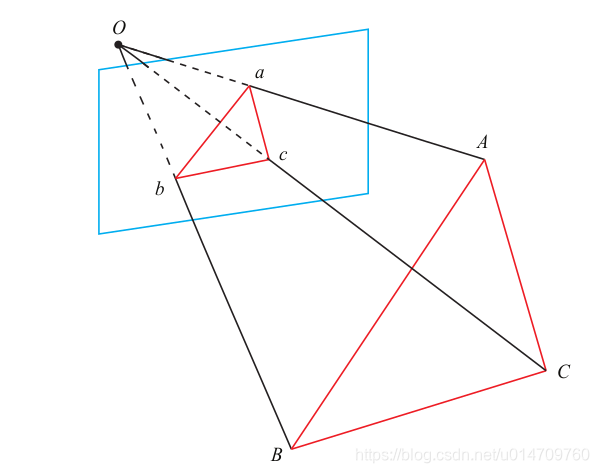
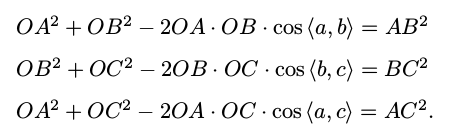
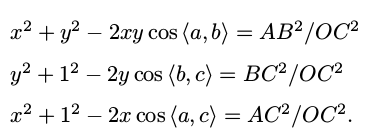
参考资料：[https://blog.csdn.net/u014709760/article/details/88029841?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522172239288016800186576237%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request\_id=172239288016800186576237&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~first\_rank\_ecpm\_v1~hot\_rank-17-88029841-null-null.142^v100^pc\_search\_result\_base1&utm\_term=PNP%E4%BD%8D%E5%A7%BF%E7%AE%97%E6%B3%95&spm=1018.2226.3001.4187](https://blog.csdn.net/u014709760/article/details/88029841?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522172239288016800186576237%2522%252C%2522scm%2522%253A%252220140713.130102334.pc%255Fall.%2522%257D&request_id=172239288016800186576237&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~first_rank_ecpm_v1~hot_rank-17-88029841-null-null.142%5ev100%5epc_search_result_base1&utm_term=PNP%E4%BD%8D%E5%A7%BF%E7%AE%97%E6%B3%95&spm=1018.2226.3001.4187)

1. PNP问题是什么：PnP问题就是在已知世界坐标系下N个空间点的真实坐标以及这些空间点在图像上的投影，如何计算相机所在的位姿（已知量是空间点的真实坐标和图像坐标，未知量（求解量）是相机的位姿）
2. 解决方法：三对点估计位姿的 P3P 、直接线性变换(DLT)、EPnP，以及非线性优化的方式，构建最小二乘问题并迭代求解（Bundle Adjustment）
3. P3P：
   1. 理解：三个点可求出4个解，其中一个是真解
   2. 数学推导：

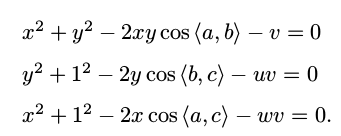


由上可得

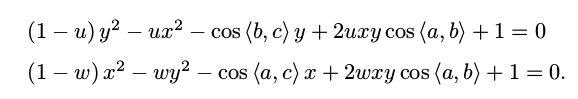
对上面三式全体除以 OC^2 ，并且记 x = OA/OC， y = OB/OC，得



记 v = AB^2 /OC ^2 , uv = BC^2 /OC^2 , wv = AC^2 /OC^2 ,有:



我们可以把第一个式子中的 v 放到等式一边,并代入第 2,3 两式,得:



由于我们知道 2D 点的图像位置,三个余弦角cos ⟨a, b⟩ , cos ⟨b, c⟩ , cos ⟨a, c⟩是已知的。同时,u = BC^2 /AB^2 , w = AC^2 /AB^2 可以通过A, B, C 在世界坐标系下的坐标算出,变换到相机坐标系下之后,并不改变这个比值，所以也是已知量。该式中的 x, y 是未知的,随着相机移动会发生变化。

因此，**P3P问题最终可以转换成关于 x, y 的一个二元二次方程(多项式方程)**

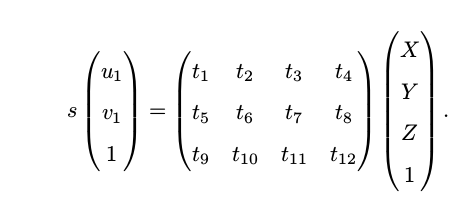
* 1. 算法的缺陷：P3P 只利用三个点的信息。当给定的配对点多于 3 组时,难以利用更多的信息，如果 3D 点或 2D 点受噪声影响,或者存在误匹配,则算法失效

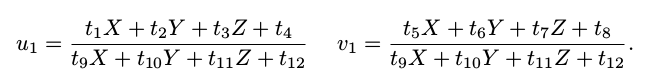
所以后续人们还提出了许多别的方法，如 EPnP、 UPnP 等。它们利用更多的信息，而且用迭代的方式对相机位姿进行优化,以尽可能地消除噪声的影响

1. 直接线性变换（DLT）
   1. 齐次坐标：在常规维度的基础上添加一个额外的维度

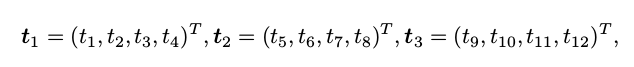
其特点是：**变换的便捷性**：齐次坐标使得包括平移、旋转、缩放和透视在内的各种变换可以统一地通过矩阵乘法来处理。这在计算机图形学中尤其有用，因为它简化了变换的计算；**表示无穷远点**：在齐次坐标系统中，可以用有限的坐标表示无穷远的点。当 w 为 0 时，表示的是无穷远的点。这在项目几何中非常重要；**从齐次坐标到非齐次坐标的转换**：要将齐次坐标转换回非齐次坐标，只需将每个坐标除以 w。例如，齐次坐标 (x, y, z, w) 对应的非齐次坐标是 (x/w, y/w, z/w)

* 1. 考虑某个空间点 P ,它的齐次坐标为 P = (X, Y, Z, 1)^T 。在图像 I 中,投影到特征点 x 1 = (u 1 , v 1 , 1)^T (以归一化平面齐次坐标表示)。此时相机的位姿 R, t 是未知的。与单应矩阵的求解类似,我们定义增广矩阵 [R|t] 为一个 3 × 4 的矩阵,包含了旋转与平移信息 。我们把它的展开形式列写如下:

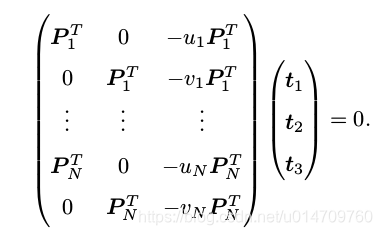


从而得到

为了简化表示,定义 T 的行向量:



于是有: 在这里插入图片描述和在这里插入图片描述

请注意 t 是待求的变量,可以看到每个特征点提供了两个关于 t 的线性约束。假设一共有 N 个特征点,可以列出线性方程组：

由于 T 一共有 12 维，因此**最少通过六对匹配点**，即可实现矩阵 T 的线性求解，这种方法(也)称为直接线性变换(Direct Linear Transform, DLT)。当匹配点大于六对时，可以使用 SVD 等方法对超定方程求最小二乘解。

在 DLT 求解中，我们直接将 T 矩阵看成了 12 个未知数，忽略了它们之间的联系。因为旋转矩阵 R ∈ SO(3)，用 DLT 求出的解不一定满足该约束，它是一个一般矩阵。平移向量比较好办，它属于向量空间。对于旋转矩阵 R，我们必须针对 DLT 估计的 T 的左边3 × 3 的矩阵块，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 [3, 48]，相当于把结果从矩阵空间重新投影到 SE(3) 流形上,转换成旋转和平移两部分。

需要解释的是，我们这里的 x 1 使用了归一化平面坐标，去掉了内参矩阵 K 的影响——这是因为内参 K 在 SLAM 中通常假设为已知。如果内参未知，那么我们也能用 PnP去估计 K, R, t 三个量。然而由于未知量的增多，效果会差一些。