Homography 知多少?

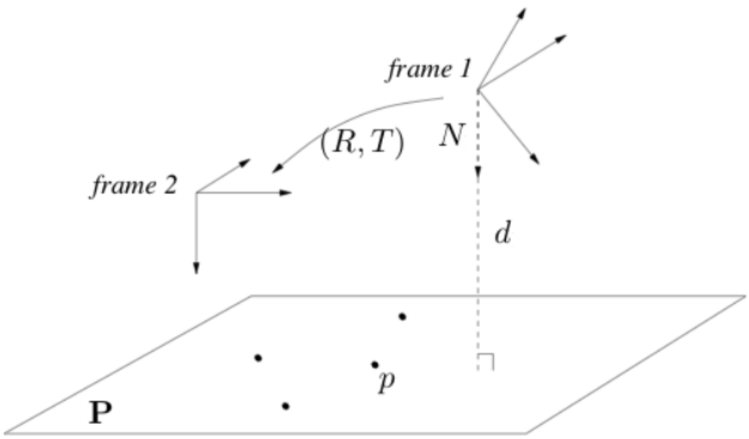
在ORB-SLAM初始化的时候，作者提到，如果场景是平面，或者近似平面，或者低视差时，我们能应用单应性矩阵(homography)，这三种情形在我应用SVO的过程中颇有同感，打破了我对H矩阵的固有映像，即只能用于平面或近似平面。但是我不知道如何去具体分析这里面的误差，比如不共面的情况时，应用H矩阵将一个图像坐标从图像1投影到图像2时，它会落在图像哪个位置？和实际位置的误差该怎么计算？误差会有多大？和哪些因素有关？另外，为何相机只做纯旋转运动时，不管平面还是非平面，H矩阵都能应用？等等，一些列问题，让我感觉对homography了解很粗浅。

先简单回顾我脑海里的H矩阵，让大家有点代入感，原谅我的啰嗦，进入正文以后就会尽量言简意赅。在没做视觉SLAM以前，通过opencv大概知道：利用两个图像中至少四个特征点能够求解一个单应性矩阵(homography matrix)，然后用这个单应性矩阵H能够将图像1中的某个坐标(u, v)变换到图像2中对应的位置(u', v')。然而，那时忽略了两个图像能够计算H的前提条件。在学SLAM过程中，知道H矩阵的推导是来自于相机在不同位姿拍摄**同一个三维平面**，所以使用opencv计算单应性矩阵H的时候前提是两个图像对应区域必须是同一平面。

最近，刘浩敏师兄的RKSLAM里面用了多H矩阵来提高鲁棒性，以及加上开头的那些疑问让我有迫切进一步学习H矩阵的想法。本文将包括三部分：H的由来，H矩阵的扩展：相机的纯旋转和非共面情形，由H矩阵到6点法估计本征矩阵E。

H矩阵的由来

假设相机在两个不同位姿处拍摄一个平面，该平面在frame 1中的法向量为N，到frame 1原点距离为d，具体如下图所示



于是，坐标系1中的点可以用下式转换到坐标系2中：

$$\mathbf{X}_2 = R\mathbf{X}_1 + \mathbf{T}$$

注意，**大写粗体$\mathbf{X}$**表示的是**三维空间点。**同时，由于三维点$\mathbf{X}_1$所在平面上，由简单的直角三角形，可知该点沿着法线方向的投影距离应等于d：

$$\mathbf{N}^T\mathbf{X}_1 = n_1X + n_2Y + n_3Z = d$$

或者

$$\frac{1}{d}\mathbf{N}^T\mathbf{X}_1 = 1 \qquad \forall \mathbf{X}_1 \in P$$

结合起来我们能够得到：

$$\mathbf{X}_2 = R\mathbf{X}_1 + \mathbf{T}\frac{1}{d}\mathbf{N}^T\mathbf{X}_1 = H\mathbf{X}_1$$

所以我们就得到了平面单应性矩阵

$$H = R + \mathbf{T}\frac{1}{d}\mathbf{N}^T, \quad H \in \mathbb{R}^{3\times 3}$$

回忆之前提到过本征矩阵$\boldsymbol{x}_2^T E\boldsymbol{x}_1 = \boldsymbol{x}_2^T \hat{T} R\boldsymbol{x}_1 = 0$，它只是把点对应到一条极线，而**单应性矩阵约束更强，是点到点的一一对应。**

注意，本征矩阵约束公式是对于归一化图像平面坐标$\boldsymbol{x} = (x, y, 1)^T$而言的，而上述推导的$H$是对三维空间点的。从3d到2d, 只需要将3d点向归一化图像平面$z = 1$上投影。三维空间点到归一化图像平面只是对坐标缩放了z，有：

$$\lambda_1\boldsymbol{x}_1 = \mathbf{X}_1, \quad \lambda_2\boldsymbol{x}_2 = \mathbf{X}_2 \to \lambda_2\boldsymbol{x}_2 = H\lambda_1\boldsymbol{x}_1$$

从这里我们可以发现，从归一化图像平面坐标$\boldsymbol{x}_2$到$H\boldsymbol{x}_1$之间还存在一个尺度因子，因此我们利用两个图像对应的坐标对能恢复H，但从该H中无法将平移$\mathbf{T}$和d分离出来，就导致了尺度的不确定性。而利用H，我们能得到$\boldsymbol{x}_2 \sim H\boldsymbol{x}_1$，注意虽然这里是用的相似符号，但是我们还是能得到图像坐标的一一对应，计算出$\boldsymbol{x} = H\boldsymbol{x}_1$以后，将$\boldsymbol{x}$的坐标都除以$\boldsymbol{x}_z$进行坐标归一化，就能得到$\boldsymbol{x}_2$。

H矩阵的扩展：相机的纯旋转和非共面情形

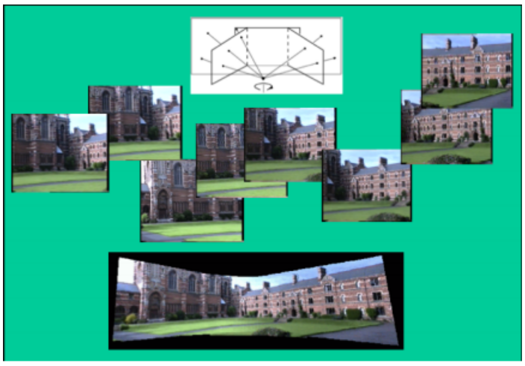
先看纯旋转情形，三维坐标关系如下：

$$\mathbf{X}_2 = R\mathbf{X}_1$$

对应的有

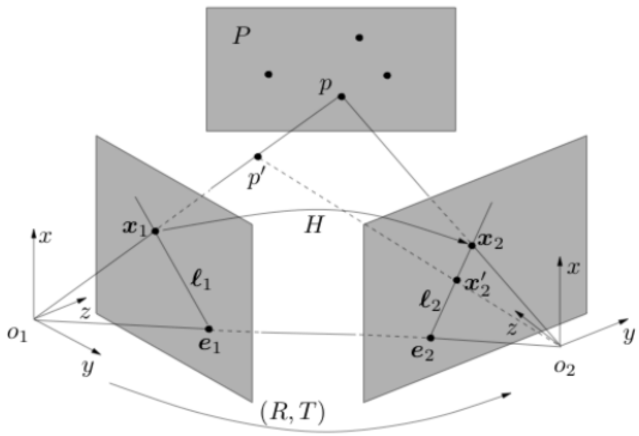
$$H = R + \mathbf{T}\frac{1}{d}\mathbf{N}^T, \quad \mathbf{T} = \mathbf{0}$$

我们可以发现公式H和深度d没有关系了，无论三维点是否在一个平面上，H矩阵都能完美的符合他们之前的转换关系。同时，平移向量为0，可以等价于所有点位于无穷远平面上，即$d \leftarrow \infty$。从另一个角度来看，如果位移为0，也无法计算d，d为任意值都能满足上面点的转换关系。所以，**大家在做全景拼接的时候，要尽量只用纯旋转哦！**当然，如果相机全景拼接算法好，就当我没说。



好了，让我们回到另一个问题，即非共面情形，又不是纯旋转，那我们使用一个H矩阵来进行坐标点的转换误差会咋样？

假设我们有一些非共面点，通过RANSAC算法估计了一个满足大多数点对应关系的H矩阵，那么对于不在三维平面上的p'用矩阵H转换以后，H会强迫它落到三维平面上，然后投影到另一个图像归一化平面，示意图如下：

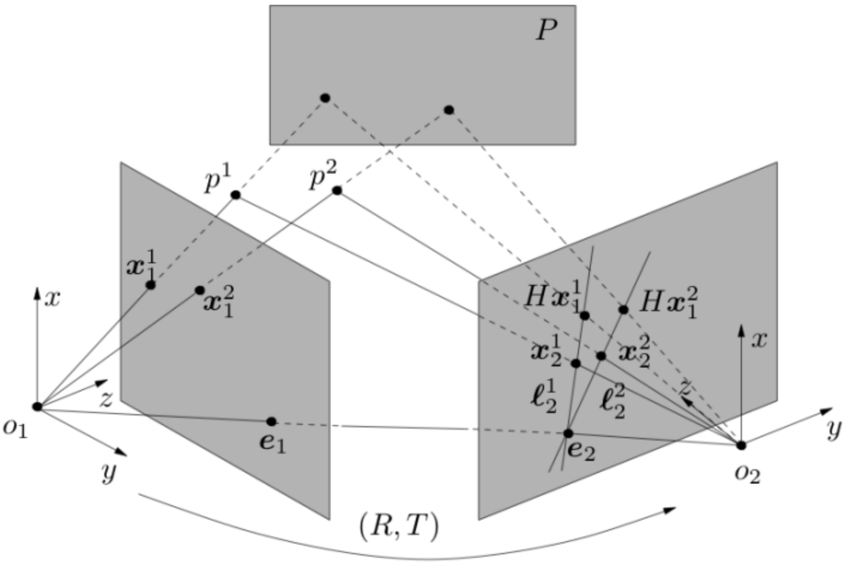


也就是说p'实际应该对应$\boldsymbol{x}'_2$，由于使用了错误的模型H，它会落到$\boldsymbol{x}_2$。$\boldsymbol{x}_2$的横纵坐标通过H矩阵能够算出，$\boldsymbol{x}'_2$坐标通过将p'进行旋转矩阵R和平移$\mathbf{T}$，再投影以后也能算出。因此两个坐标相减就能得到误差模型。不需要精确的数学计算，单单从上面的图，我们就能直观的感受到**当相机的平移向量相对于场景深度而言足够小时，$\boldsymbol{x}'_2$和$\boldsymbol{x}_2$之间的误差是可被接受的，即这种情况下依然可以使用H矩阵来算坐标的一一对应**，这应该就是orbslam中提到的低视差情形。

一个重要的启示是：H矩阵的这种直接计算图像坐标一一对应关系的性质给解决SLAM中像素点的匹配又提供了一条思路，如果一个H矩阵不行，那一幅图片就用多个H矩阵，这正是浩敏师兄那篇ISMAR 2016论文的重要框架基础。而通过本征矩阵E只能计算极线，还需要沿着极线匹配。

由H矩阵到本征矩阵E的一点遐想

既然说到了极线，顺着上面的思路，干脆探一探本征矩阵E。其实，上面图中，大家伙都看到$\boldsymbol{x}'_2$和$\boldsymbol{x}_2$是位于极线上的，他们坐标都知道，就能得到极线方程，如果还有另外一个点再确定一条极线，如下图所示



那么外极点e_2就能确定，极点确定了，H也知道，那么其他任意点的极线就能画出来了，不用本征矩阵我们也可以构造极线几何。这就是所谓的“六点法”，四个共面点确定H,两个非共面点确定极点。

另外，还可以解释为什么8点法计算求解本征矩阵不能应用于共面的情形。我们知道一个向量和自己叉乘结果等于0，所以有

$$\hat{x}_2 \boldsymbol{x}_2 = 0 \rightarrow \hat{x}_2 H \boldsymbol{x}_1 = 0$$

另外，任意三维向量$\boldsymbol{u}$和$\boldsymbol{x}_2$的叉乘肯定垂直于向量$\boldsymbol{x}_2$，所以有：

$$\begin{gathered} \hat{u}\boldsymbol{x}_2 \perp H\boldsymbol{x}_1 \\ \Rightarrow (\hat{u}\boldsymbol{x}_2)^T H \boldsymbol{x}_1 = 0 \\ \Rightarrow -\boldsymbol{x}_2^T \hat{u} H \boldsymbol{x}_1 = 0 \\ \Rightarrow \boldsymbol{x}_2^T \hat{u} H \boldsymbol{x}_1 = 0 \end{gathered}$$

所以我们能得到$E = \hat{u}H$，这说明E在这种情况下有无穷多解都能满足$\boldsymbol{x}_2^T E \boldsymbol{x}_1 = 0$，这个时候还用8点法，肯定是不行了。

总结

总算写完了，推荐大家去读Yi Ma的书《An Invitation to 3D vision》上面几乎都来自于这本书，同时之前也提到过TUM的Prof.Cremers上课就是用的这本书。除此之外，再推荐个基于该书的课程。视觉几何真心水深，没有理论积累，没有前人指点，坑是填不完的，祝好(对大家，也是对自己)。