CubeSLAM论文分享

- 论文链接：CubeSLAM_Monocular_3-D_Object_SLAM.pdf - 飞书云文档 (feishu.cn)
- 开源：https://github.com/shichaoy/cube_slam
- 笔记：Dynamic SLAM总结.pdf - 飞书云文档 (feishu.cn) CubeSLAM注记 - 飞书云文档 (feishu.cn)

CubeSLAM在解决什么问题

传统的经典SLAM，如单目ORB-SLAM存在以下的问题：

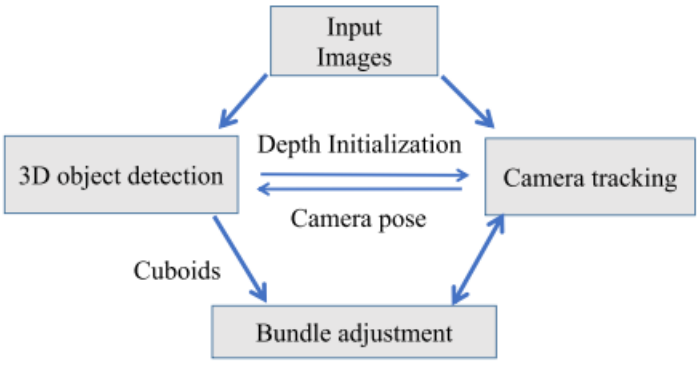
1. 单目SLAM固有的尺度不确定性带来的精度损失

   单目SLAM的初始化过程(主要为三角化)本身就带有误差，尤其是在相机只发生旋转或者平移量较小时，并且误差会随时间累积产生尺度漂移。经典的ORB-SLAM主要通过回环检测来减少漂移，但在实际情景中，回环的出现可遇不可求。

2. 基于点的SLAM方法不够适应复杂的动态环境

   在追踪相机时，主流的方法之一是通过特征点匹配来估计位姿。实际情景下，动态特征点的匹配本身就有困难，而当特征点来自动态物体时，会带来不准确的估计。虽然可以通过对极约束来对静态点与动态点进行区分，但浪费了物体本身的运动信息。 另外，如遇过大相机旋转等情形，匹配的特征点数量较少，基于点的SLAM方法可能会失效。

CubeSLAM作为单目的物体级SLAM，保留了ORB-SLAM的特征提取与关键帧创建，并在其基础之上引入了**物体信息**，放入BA优化过程中，以更好的适应实际情景。流程图如下：



(a)

我们得到的3D物体信息本身依赖于输入的图像与相机的位姿，为相机的追踪提供深度初始化，并为BA过程提供物体信息带来的误差函数。此外，我们可以基于物体信息挑战特征点过少的情形。

如何得到物体信息(3D object detection)

本文通过最简单的立方体来表示物体信息(给实际物体3D的立方体外框)，立方体的信息可由9个参数决定，包括位移量$t = [t_x, t_y, t_z]$，旋转量$R \in SO(3)$，与三条边的长度$d = [d_x, d_y, d_z]$。以上参数都为中心为原点的立方体坐标系相对于世界坐标系的值。我们将基于深度网络提供的bounding box,灭点(VP)点以及相机位姿来确定立方体。

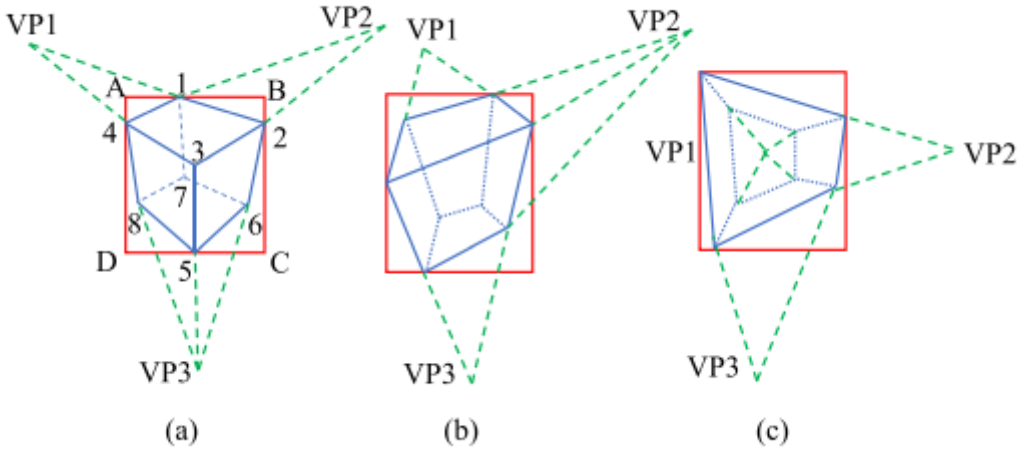
如何得到立方体的2D信息

首先我们需要目标检测来给出图像平面上的bbox。作者在室内数据集使用的是YOLO，在KITTI数据集上使用的是MS-CNN，总之我们要得到如下的2D bbox。

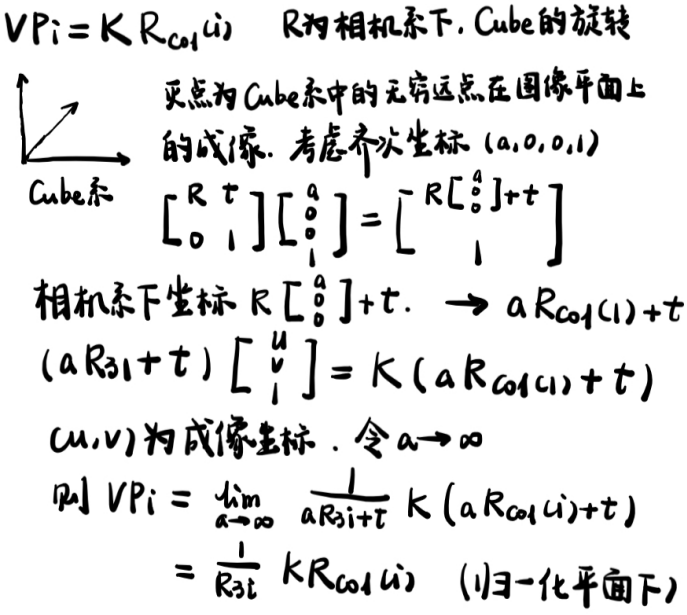
Final detections

同时我们做出假设：物体真实的立方体边框的顶点落在bbox的边上。根据此假设，bbox的四条边可以为立方体信息带来四个约束条件，但显然不足以确定整个立方体边框。文章创新性地引入了灭点(VP点)来给出立方体信息。

立方体成像后根据看到面的数量只有三种情况。对于三种情况，确定三个VP点与一个顶点的2D坐标就可以确定8个点的2D坐标。



VP点是二维图像中三维平行线的交点，具有这样的特殊性质：其坐标只与立方体坐标系相对于相机坐标系的旋转矩阵有关。其计算公式为$VP_i = KR_{col(i)}, i \in 1,2,3$，$R_{col(i)}$表示矩阵R的第i列。



由上所述，确定立方体的2D坐标变成了确定立方体坐标系相对于相机坐标系的旋转矩阵R与其中某一顶点的2D坐标。由于这两个量都不好直接计算确定，我们引入采样和评估的办法，这在后面会详细讨论。我们先假定立方体8个顶点的2D坐标已知，来复原其3D坐标。

如何得到立方体的3D信息(相对于相机坐标系的3D坐标)

对于一般物体而言，立方体八个顶点在立方体参考系下的坐标为$[\pm d_x, \pm d_y, \pm d_z]/2$，在相机参考系下的坐标为$R[\pm d_x, \pm d_y, \pm d_z]/2 + t$。对于某一顶点$p1$，我们可以建立投影方程$p_1 = \pi(R[d_x, d_y, d_z]/2 + t)$，$\pi$是相机决定的投影函数。$p_1$是二维坐标，故提供两个约束。而对于单目情形下的单帧图像，不方便直接确定一般物体的尺度，因而对于除去尺度外的8个自由度，需要四个顶点就可以求解。

对于地面上的物体，有更简单的求解办法并且可以轻易地得到尺度量。我们将立方体顶点的像素点反投影得到一支三维射线(射线上的所有点成像坐标相同)。射线于地平面的交点就是对应的立方体顶点，尺度量由相机距地面的高度所决定。记相机参考系下，$[\mathbf{n}, m]$分别为地面的(单位)法向向量与相机距离(需要用到相机的位姿信息)，p_1为二维像素点坐标，P_1为三维对应坐标。做如下推导：

设直线过(m_1, m_2, m_3),方向向量为(v_1, v_2, v_3)，平面过点(n_1, n_2, n_3),法线为$(vp_1，vp_2，vp_3)$,
交点坐标为$x = m_1 + v_1 t$,$y = m_2 + v_2 t$,$z = m_3 + v_3 t$,
其中$t = \frac{((n_1-m_1)vp_1+(n_2-m_2)vp_2+(n_3-m_3)vp_3)}{(vp_1v_1+vp_2v_2+vp_3v_3)}$.
在我们的问题中，选取相机参考系，射线过$(0,0,0)$,方向向量为$K^{-1}p_5$,平面法向为$\mathbf{n}$，设平面经过点$(0,0,a)$,则根据距离m求得$a = -m/n_3, \mathbf{n} = [n_1, n_2, n_3]$,从而$P_1 = -\frac{m}{\mathbf{n}^T(K^{-1}p_5)}K^{-1}p_1$.

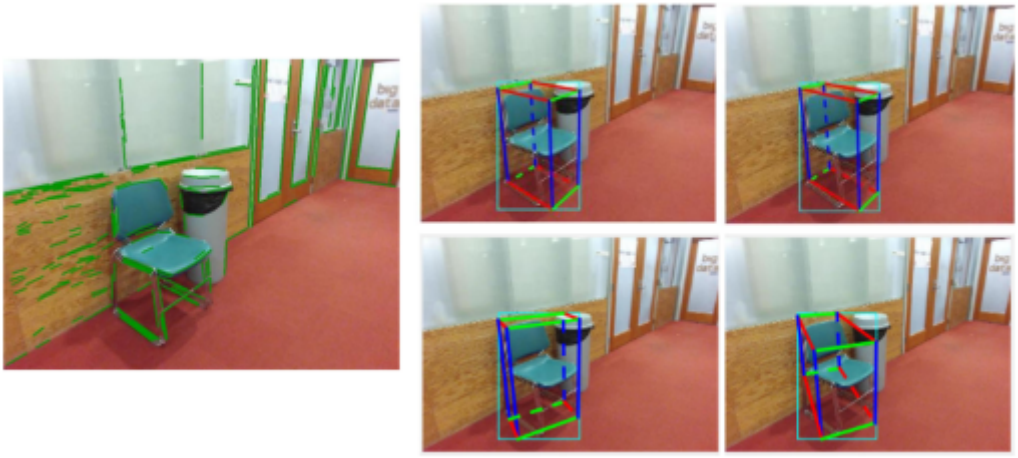
利用这样的办法我们可以得到8个顶点的3D坐标，这与立方体假设是分不开的。

采样与评估

现在我们已经可以通过先前提到的旋转矩阵与某一顶点的2D坐标计算立方体八个顶点的3D坐标。

对于一般的物体，我们需要对整个旋转矩阵采样。对于地面上的物体，采样所用到的是立方体的yaw角，相机的roll角和pitch角(我们可以建立特定的世界坐标系，使得这些量决定旋转矩阵R,考虑将相机的z轴与地面法向平行)。对于某些数据集，如KITTI等，相机的roll与pitch都是提供好的。对于多视图的视频数据，可以通过SLAM给出相机位姿的一个初值，并在此附近采样以减少采样空间。

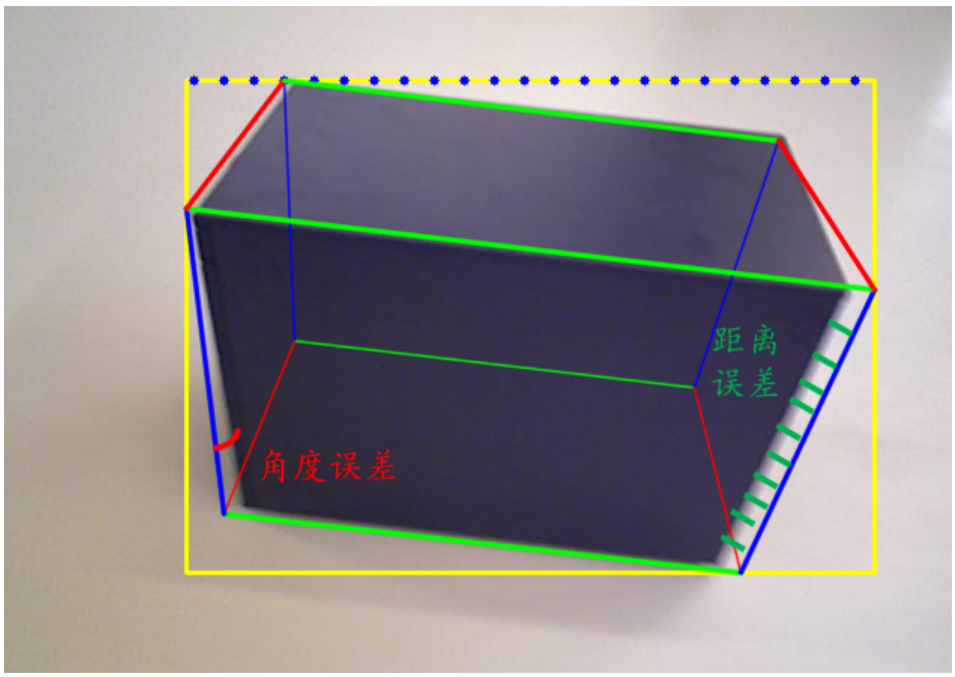
为确定立方体的一个顶点，我们对bbox的上边缘进行等距采样。顶点的采样与先前的旋转矩阵采样两两进行组合，可以得到一些立方体的提案。我们需要在此之中选择最贴合物体的立方体提案，从而引入提案的评价机制。



在采样了一些立方体提案后，我们通过定义损失函数来为他们打分，主要考虑VP点与边缘等约束以及一定的先验信息，以下的方法对"具有明显边缘的箱体"具有很好的效果。记I为图像，立方体提案记为$O = \{R, t, d\}$，损失函数表示为：

$$E(O|I) = \phi_{dist}(O, I) + \omega_1\phi_{angle}(O, I) + \omega_2\phi_{shape}(O, I) \tag{17}$$

其中$\phi_{dist}, \phi_{angle}, \phi_{shape}$为三种不同的损失，$\omega_1, \omega_2$表示权重，作者在数据集测试时分别设置为0.8和1.5。

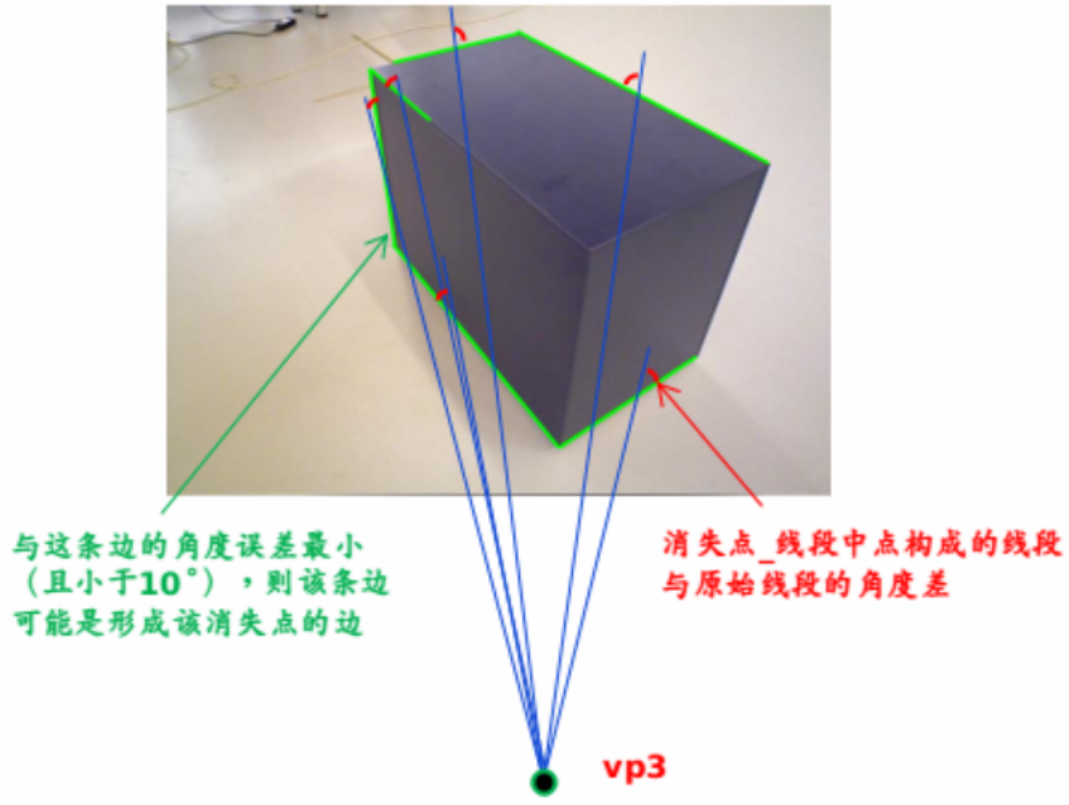


1. 距离误差$\phi_{dist}(O, I)$: 二维成像下立方体的边缘应当尽可能与图像中的边缘尽可能匹配

   检测Canny边缘，并根据他们在图像上做distance transform(DT)变换。在所有可见边(立方体提案二维成像的边)上等距采样并求和，再除以2D bbox的对角线长。距离误差对假正例(FP)非常敏感，比如物体表面纹理。

2. 角度误差$\phi_{angle}(O, I)$：线检测得到的边应当与VP点计算得到的边尽可能匹配

   通过线检测处理得到长线条以后，首先我们需要找出形成消失点的边。主要思想是计算消失点-线段中点所构成线段的角度，与检测到的线段的角度进行比较，如果角度差在阈值内，则该条线段可能是形成消失点的线段（理论上消失点和所支持的线段的中点是在一条线上的）。



为了保证一定的泛化性能，如果检测到多条满足角度要求的边，则选择角度最大和最小的，作为形成消失点的两条“支撑线”，分别记为$< l_{i_ms}, l_{i_mt} >$,$< l_{i_ns}, l_{i_nt} >$，$l_{i_ms}, l_{i_mt}, l_{i_ms}, l_{i_mt}$分别为两条边对应的顶点，尖括号表示两个顶点形成的边。角度误差记为

$$\begin{aligned}\phi_{angle}(O, I) = \Sigma_{i=1:3} \| < l_{i\_ms}, l_{i\_mt} > - < VP_i, l_{i\_mt} > \| + \\ \| < l_{i\_ns}, l_{i\_nt} > - < VP_i, l_{i\_nt} > \|\end{aligned} \tag{18}$$

3. 形状误差：惩罚高长宽比的立方体提案

   先前两种误差都是服务于2D图像空间，但相似的2D顶点可能产生完全不同的立方体。文中为此提出了形状误差，其中$s = max(d_x/d_y, d_y/d_x)$。

$$\phi_{shape}(O) = max(s - \sigma, 0) \tag{19}$$

   当$s < \sigma$时，不进行任何惩罚，σ为预设的阈值。

   实际上我们可以利用更严格的先验来得到更有效的形状误差。

确定立方体提案

根据采样得到的立方体提案与评价机制，我们选择最好的立方体提案作为最终的结果，提供深度初始化以及进入BA。提案数据包括立方体2D坐标，3D坐标，相应计算得到的9个参数。

物体信息如何在SLAM中起到作用

我们将3D立方体的检测信息放入SLAM中，修改ORB-SLAM的BA模块，联合优化物体位姿与相机位姿。

静态物体数据关联(先物体关联还是先点关联？)

静态物体的数据关联，包括不同帧间的点-点、同一帧内的点-物体以及不同帧间的物体-物体。

不同帧间点与点的数据关联主要通过描述子匹配与对极检查。此处主要关联静态特征点，因而采用对极检查滤除来自动态物体的特征点。

同一帧内点与物体的数据关联主要通过2D与3D检测。如果同一点在至少两帧在2D bbox内被观测到，并且距离立方体3D中心的距离小于1m，则将特征点关联到对应的物体。

不同帧间物体与物体的数据关联主要通过点的关联来考虑。如果不同帧的两物体共享一定数量的特征点，并且数量超过一定阈值，则在不同帧间关联两个物体。

另外，如果某物体关联的特征点数量过少，则认为物体是动态的，将被丢弃。

针对静态问题的BA

我们在BA中引入相机位姿，点与物体位姿，分别记为$C = \{C_i\}, O = \{O_j\}, P = \{P_k\}$，考虑如下非线性优化：

$$C^\star, O^\star, P^\star = \underset{C,O,P}{argmin} \sum_{C_i, O_j, P_k} \|e(c_i, o_j)\|^2_{\Sigma_{ij}} + \|e(c_i, p_k)\|^2_{\Sigma_{ik}} + \|e(o_j, p_k)\|^2_{\Sigma_{jk}} \tag{20}$$

其中$e(c,o)$,$e(c,p)$,$e(o,p)$分别代表相机-物体，相机-点，物体-点的测量误差,Σ代表不同测量的的协方差矩阵。

符号约定：$T_c \in SE(3)$表示相机位姿，点坐标表示为$P \in \mathbb{R}^3$，物体将由9参数建模,$O = \{T_o, \mathbf{d}\}$，$T_o = [R \quad t] \in SE(3)$为位姿表示，$\mathbf{d} \in \mathbb{R}^3$为物体的三个维度。在有些数据集如KITTI中，$\mathbf{d}$是给好的参数，不需要加入BA中。

1. 相机-物体测量：
   - 3D测量误差：

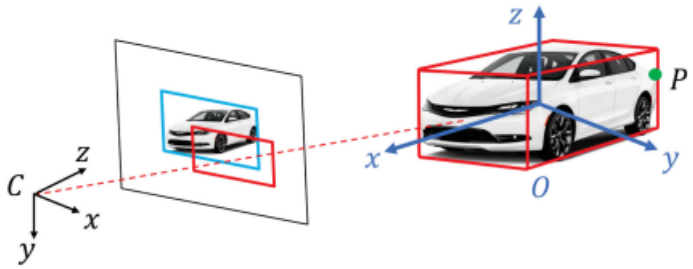
     3D测量误差需要3D信息较为准确时才能使用，如使用RGB-D的情形(但我们更多考虑单目)。设单张图片检测到物体的位姿与三维信息为$O_m = (T_{om}, \mathbf{d}_m)$，我们将它转换到相机系下，与测量比较，得到误差

     $$e_{co\_3D} = [log((T_c^{-1}T_o)T_{om}^{-1})^{\vee}_{\mathbf{se}_3} \quad \mathbf{d} - \mathbf{d}_m] \tag{21}$$

     注意到前一项的误差越小，$(T_c^{-1}T_o)T_{om}^{-1}$越接近于单位矩阵，我们需要引入log与$\vee$来将$SE(3)$空间映到$\mathbb{R}^6$来实现最优情况在数值上最小(接近于0)。所有测量误差都会引入Huber核来提高鲁棒性。

     立方体提案生成无法确定背面与正面，相差一个旋转的相同立方体可能被认为是不同的立方体。从而上述误差计算时，我们对立方体进行0°， $\pm 90^\circ, 180^\circ$的旋转并取最小误差，作为3D测量误差。
   - 2D测量误差：

     而对于单目的情形，更多的还是考虑2D测量误差。我们将立方体路标投影到图像平面，得到2D长方形。

     如上图，红色长方形为立方体提案在图像平面上的投影，蓝色长方形为目标检测给出的bbox，我们将根据两个长方形的匹配度来计算2D测量误差，红色长方形将由8个顶点对应投影点的坐标关系给出，考虑：

     $$[u,v]_{min} = min\{\pi(R([\pm d_x, \pm d_y, \pm d_z]/2 + t))\} \tag{22}$$

     $$[u,v]_{max} = max\{\pi(R([\pm d_x, \pm d_y, \pm d_z]/2 + t))\} \tag{23}$$

     $$\mathbf{c} = ([u,v]_{min} + [u,v]_{max})/2 \quad (\text{中心对应坐标}) \tag{24}$$

     $$\mathbf{s} = [u,v]_{max} - [u,v]_{min} \quad (\text{对角线向量}) \tag{25}$$

     我们比较$\mathbf{c}, \mathbf{s}$在测量长方形与检测bbox上的差值$e_{co_2D} = [\mathbf{c}, \mathbf{s}] - [\mathbf{c}_m, \mathbf{s}_m]$，作为2D测量误差。

     2D测量相对于3D测量的不确定性更小，在于2D信息相比3D信息更加准确，而对于单目情况，3D信息的不确定性更大。但不同的3D立方体可能投影到相同的2D长方形，从而我们需要更多的约束。

2. 物体-点测量：

   物体与点可以根据静态数据关联来彼此约束。如果某点关联到物体，则该点应落在物体的立方体边框内。所以我们将点坐标变换到立方体坐标系，并与立方体的三维比较得到物体-点测量误差：

   $$e_{op} = max(|T_o^{-1}P| - \mathbf{d}_m, \mathbf{0}) \tag{26}$$

3. 相机-点测量：

我们使用标准的重投影误差，其中z_m是3D点P被观测到的的像素坐标：

$$e_{cp} = \pi(T_c^{-1}P) - z_m \tag{27}$$

当匹配特征点数量过少时，ORB-SLAM可能完全失效。而CubeSLAM的BA中，相机-物体测量还得到保留，具有挑战这种复杂环境的能力。

动态环境数据关联

作者通过实验后认为静态环境下的关联方法可能不适于解决动态问题，原因在于动态特征点的匹配上有一定的困难。解决动态特征点匹配的典型方法是预测匹配点的位置，在其附近搜索。对于单目动态情形，精确估计物体的运动比较困难。我们对点-物体的关联问题提出了不同的方法，基于点追踪和动态目标三角化(可以参考文档动态目标三角测距方法 - 飞书云文档 (feishu.cn))。

首先我们使用2D KLT稀疏光流追踪替代特征匹配来进行点追踪，同时不需要点的3D信息。在像素点追踪后，考虑动态特征点的三角化问题来恢复点的3D坐标。假设两帧对应的投影矩阵分别为M_1, M_2，两帧的3D点坐标分别为P_1, P_2，$P_2 = \Delta T P_1$，ΔT为两帧间物体的变换矩阵，两点对应的像素坐标分别为z_1, z_2，则有

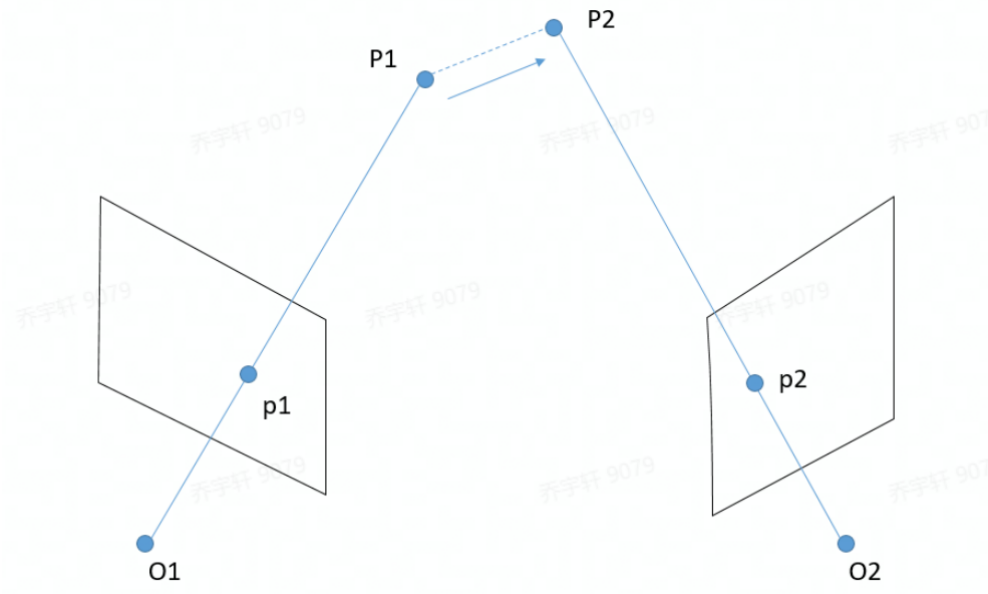
$$M_1 P_1 = z_1 \tag{28}$$

$$M_2 \Delta T P_1 = z_2 \tag{29}$$

若将$M_2\Delta T$视为针对物体运动补偿所得的投影矩阵，则问题变为了标准的静态三角化问题。注意M_1, M_2中包括了相机的位姿变化与P_1, P_2的深度，实际上两式与动态目标三角测距方法 - 飞书云文档 (feishu.cn)中的对极问题

$$s_2 p_2 = s_1 K T_{o21} \Delta T K^1 p_1$$

是完全等价的(符号定义有所不同)，我们只需要根据相机的运动就可以确定物体的运动。



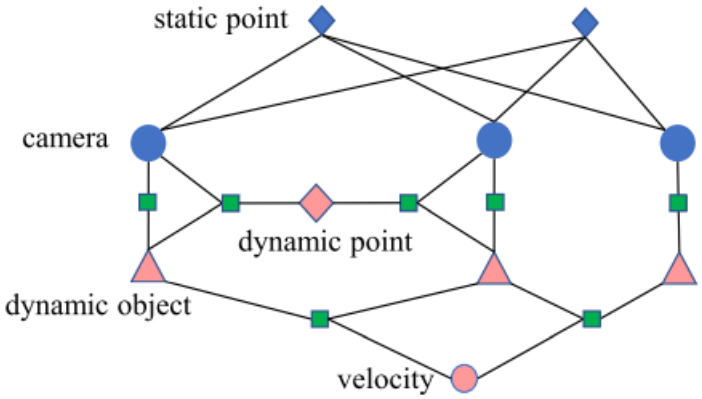
引入KLT稀疏光流追踪与动态问题的三角化我们可以确定特征点的3D坐标，进一步进行点-物体关联?

KLT追踪在像素位移量过大时可能失败，从而我们对于动态问题物体-物体关联不使用静态的共享特征点来判断。我们使用文献[40]的办法直接对2D bbox进行追踪，预测以及匹配。由此实现物体-物体的关联。

针对动态环境的BA

在这部分，我们提出一种办法联合优化物体运动与相机位姿。首先做出如下两个假设：刚体假设与匀速假设。记${}^jO^i$为动态物体O^i在j帧观测到的位姿。对O^i上的动态点P^k，我们用立方体系下的坐标${}^iP^k$表示，注意由于刚体假设${}^iP^k$不随时间而变化，也不能直接用于SLAM优化中(需要先坐标变换)。

动态环境下的BA添加了两个新的动态因子，包括物体运动因子与点-相机-物体因子。物体-相机因子与静态相同。



1. 物体运动模型：

   我们将动态物体视为汽车进行运动学建模。汽车在地面上运动，位姿只有三个自由度，分别是t_x, t_y, θ，t_x, t_y表示汽车在二维平面上的平移量，θ为汽车的yaw角。记汽车的速度为v，航向角为ϕ，则预测状态

   $$[t_x^{'}, t_y^{'}, \theta^{'}]^T = [t_x, t_y, \theta]^T + v\Delta T[\cos(\theta), \sin(\theta), \tan(\phi)/L]^T \tag{30}$$

   (14)式来自Kinematic Model，可参考链接车辆控制-运动学模型(Kinematic Model)_zgh-CSDN博客

   其中L是前轮中心与后轮中心的距离，注意此模型坐标系在后轮中心，与立方体坐标系相差$L/2$的补偿。最终物体运动模型的误差为：

   $$e_{mo} = [t_x^{'}, t_y^{'}, \theta^{'}] - [t_x, t_y, \theta] \tag{31}$$

   此处，前一项是根据t时刻的观测对$t + \Delta t$的预测，后一项是$t + \Delta t$时刻的观测。

2. 动态点测量：

   根据动态环境数据关联，已将动态点关联到相应的动态物体，记O^i在第j个图像的位姿为${}^jT_O^i$，考虑重投影：

   $$e_{dp} = \pi({}^jT_O^i \cdot {}^iP^k, T_c^j) - z_{kj} \tag{32}$$

   其中T_c^j为第j个相机位姿，而z_{kj}为对应特征点。此处$\pi({}^jT_O^i \cdot {}^iP^k, T_c^j) = \pi(T_c^{j-1} \cdot {}^jT_O^i \cdot {}^iP^k)$。

实际情景下匀速假设自然不太可能满足。作者分析了实际的物体运动，认为物体在5s左右速度保持相同。

物体信息如何提供深度信息并减少单目漂移(文中未提及)

1. 立方体的尺度信息是已知的。立方体顶点处深度完全确定，立方体内部的点深度范围确定。这些信息可以来约束三角化后的深度结果。
2. 关联好的立方体在顶点的匹配效果上可能比特征点更好。利用立方体顶点的2D-2D关系可能得到更为精确的相机位姿，减少不稳定情形(相机平移量过小)带来的影响。实际上立方体顶点的3D信息我

们也已知，是否可以用3D-3D,3D-2D的位姿求解结果呢？注意到立方体顶点的3D坐标本身来自相机位姿，这可能会带来问题。

一些问题

- 注意到相机初始位姿估计在CubeSLAM中起到了重要作用，我们立方体的3D信息非常依赖初始位姿估计。能否在相机追踪中进行一些修改提高位姿的精度来实现性能的提高，如VDO-SLAM中分离并滤去动态点、引入光流提高鲁棒。
- 如上条相似地，CubeSLAM主要贡献了3D物体模块，并着眼后端BA，能否与一些较为先进的视觉里程计，如ClusterVO进行整合，实现更好的效果。
- 如果我们期望SLAM提供速度信息，而不依赖于测速，能否依赖物体位姿信息估计速度。实际上，对于刚体而言，速度是可以由刚体的位姿变换直接计算得到的，此部分可参考Dynamic SLAM:TNFS。
- 立方体提案的打分中，没有一项能提供良好的3D约束，原因可能在于3D的约束可能对深度信息有所需求。形状的先验信息设定也许可以根据我们的需求有所改变。