VDO-SLAM

- 论文：VDO-SLAM A Visual Dynamic Object-aware.pdf
- 开源：https://github.com/halajun/vdo_slam

VDO-SLAM是我们Dynamic SLAM调研中较新颖的算法，发表于2020年。它是由相同的作者在Dynamic SLAM:The Need For Speed 的思想上的进一步实现。先前已有关于VDO-SLAM的相关调研Dynamic SLAM总结.pdf，其内容主要参考专栏https://zhuanlan.zhihu.com/p/353878061，在内容推导方面有些不够详实，故新开一文档介绍VDO-SLAM来着重细节，可与先前文档结合参考。

介绍

相比于Dynamic_SLAM_The_Need_For_Speed.pdf，VDO-SLAM不仅仅着眼于后端，它依靠更稠密的物体特征(引入光流)来保证鲁棒追踪，并且在城市驾驶环境中引入新的平滑移动因子，使得VDO-SLAM相比于先前，成为一个完整的SLAM系统。

对于动态环境下的SLAM问题，传统的方法要么是将来自动态物体的传感数据作为外点，将其在估计过程中剔除，要么是识别动态物体并用多目标追踪分别追踪动态物体。前者浪费了来自动态点的大量信息，并且只能建静态图(在自动驾驶中，这显然是不足的，我们需要动态物体的距离和速度信息)；后者十分依赖相机的位姿估计，在复杂的动态环境下比较脆弱。VDO-SLAM解决了两者的缺陷，专注于精确地估计场景中动态物体的移动。

主要贡献

VDO-SLAM是一种新颖的基于特征的双目\RGB-D动态SLAM系统，根据基于图像的语义信息来同时定位机器人，静态与动态结构，跟踪场景中的刚体运动，主要贡献有：

- 一种新颖的动态场景建模，基于机器人位姿，静态动态3D点以及物体运动的估计
- 优于当前最好算法的动态物体的精确SE(3)移动估计，与一种提取运动物体速度的方法。运动速度的信息尤为重要，在自动驾驶的碰撞避免之类的任务中可以起到重要帮助
- 利用语义信息给出一种追踪动态物体的鲁棒办法，它具有处理遮挡物出现时语义分割失败情形的能力，即动态物体被遮挡的情形下依旧能被很好的追踪
- 在复杂真实环境下的完整系统(得益于考虑因子时的全面性，如平滑移动因子)

内容细节

1. **基本公式**

   A.背景与基本符号

记 ${}^0X_k, {}^0L_k \in SE(3)$ 分别代表相机与物体在时间k的3D位姿,相对于全局参照系{0},记 ${}^0m_{k-1}^i$ 为第i个3D点在k时刻的齐次坐标(相对于参考系{0})，此时有 ${}^{X_k}m_k^i = {}^0X_k^{-1} \cdot {}^0m_k^i$.注意到在 Dynamic SLAM:The Need For Speed 中类似的讨论，仅仅只是符号上的不同，因此与其中的推导完全一致地，可以得到 ${}^0m_k^i = {}_{k-1}^0H_k \cdot {}^0m_{k-1}^i$,与Dynamic SLAM:TNFS不同的是，我们在此处要考虑重投影误差， 记I_k为相机拍下二维图像的相对参照系，${}^{I_k}p_k^i$ 为${}^{X_k}m_k^i$ 在I_k中对应的像素位置，由投影函数$\pi(\cdot)$得到，${}^{I_k}p_k^i = \pi({}^{X_k}m_k^i) = K \cdot {}^{X_k}m_k^i$，其中$K$是相机的内参矩阵 这部分内容可由下图展示，可结合Dynamic SLAM:TNFS一起理解(先前已经过细致推导)。

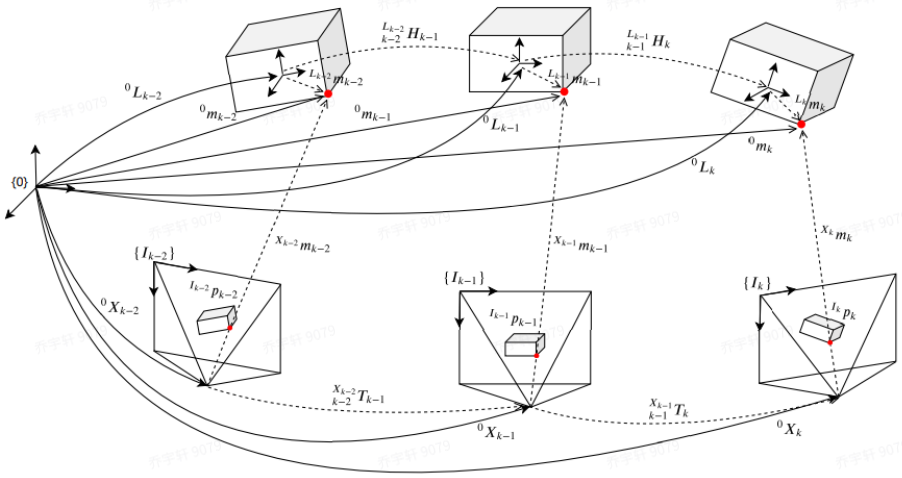


Fig. 2: **Notation and coordinate frames.** Solid curves represent camera and object poses in inertial frame; ${}^0\mathbf{X}$ and ${}^0\mathbf{L}$ respectively, and dashed curves their respective motions in body-fixed frame. Solid lines represent 3D points in inertial frame, and dashed lines represent 3D points in camera frames.

此外，我们引入光流的估计 ${}^{I_k}\phi^i = {}^{I_k}\tilde{p}_k^i - {}^{I_{k-1}}p_{k-1}^i$,其中 ${}^{I_k}\phi^i$ 是第i个点在二维图像上从k-1时刻到k时刻的光流， ${}^{I_k}\tilde{p}_k^i$ 是 ${}^{I_{k-1}}p_{k-1}^i$ 在k时刻的二维图像坐标系下的对应坐标。

B.相机位姿与运动估计

我们进行相机位姿与物体运动估计时选取的损失函数时3D-2D的重投影误差，原因在于噪声在图像平面上有更好的特征，可以导出相机定位更精确的结果。此外，我们还提出了一种创新的办法来联合优化考虑光流后的误差项来实现鲁棒追踪。

1)相机的位姿估计：相机位姿 ${}^0X_k$ 处的重投影误差为 $e_i({}^0X_k) = {}^{I_k}\tilde{p}_k^i - \pi({}^0X_k^{-1} \cdot {}^0m_{k-1}^i)$,此处选取的 ${}^0m_{k-1}^i$ 来自k-1时刻观测到的静态特征点，从而 ${}^0m_{k-1}^i = {}^0m_k^i$ ，记 $x_k \in$ se(3)为 X_k 对应的李代数，有指数映射 ${}^0X_k = exp({}^0x_k)$ ，${}^0x_k^\vee$ 为se(3)到 $\mathbb{R}^6$ 的vee operator。考虑

$$ {}^0\mathbf{x}_k^{*\vee} = \underset{{}^0\mathbf{x}_k^\vee}{\operatorname{argmin}} \sum_i^{n_b} \rho_h \left(\mathbf{e}_i^\top ({}^0\mathbf{x}_k) \Sigma_p^{-1} \mathbf{e}_i ({}^0\mathbf{x}_k) \right) \tag{9} $$

其中 ρ_h 为Huber函数，主要实现误差的鲁棒， Σ_p^{-1} 为最小二乘的协方差矩阵， n_b 为所有静态背景点的数量。相机的位姿估计由 ${}^0X_k^* = exp({}^0x_k^*)$ 给出，(9)可由L-M算法求解。(李代数将约束优化转化为无约束优化)

2)物体的移动估计：与相机位姿估计相似地，我们可以建立物体移动 ${}_{k-1}^0H_k$ 基于重投影误差的损失函数。由 ${}^0m_k^i = {}_{k-1}^0H_k \cdot {}^0m_{k-1}^i$ ， $e_i({}_{k-1}^0H_k) = {}^{I_k}\tilde{p}_k^i - \pi({}^0X_k^{-1} \cdot {}_{k-1}^0H_k \cdot {}^0m_{k-1}^i) =$

${}^{I_k}\tilde{p}_k^i - \pi({}^0_{k-1}G_k \cdot {}^0 m_{k-1}^i)$ ，其中 ${}^0_{k-1}G_k = {}^0 X_k^{-1} \cdot {}^0 m_{k-1}^i \in SE(3)$,记 ${}^0_{k-1}g_k$ 为其对应的李代数，有指数映射 ${}^0_{k-1}G_k = exp({}^0_{k-1}g_k)$,考虑以下损失函数

$$ {}^0_{k-1}\mathbf{g}_k^{*\vee} = \underset{{}^0_{k-1}\mathbf{g}_k^{\vee}}{\operatorname{argmin}} \sum_i^{n_d} \rho_h \left(\mathbf{e}_i^{\top}({}^0_{k-1}\mathbf{g}_k)\, \Sigma_p^{-1}\, \mathbf{e}_i({}^0_{k-1}\mathbf{g}_k) \right) \tag{11} $$

其中 ρ_h 为Huber函数， Σ_p^{-1} 为最小二乘的协方差矩阵， n_d 为所有动态点的数量。

3)加入光流的联合估计：相机位姿与物体运动估计都依赖好的图像相关性，这在遮挡、较大移动与相机-物体距离较大时会具有挑战性。为了保证鲁棒追踪，引入光流来联合估计。将 ${}^{I_{k-1}}\tilde{p}_{k-1}^i$ 用光流来代入1)中的误差得到新误差项 $e_i({}^0 X_k, {}^{I_k}\phi) = {}^{I_{k-1}} p_{k-1}^i + {}^{I_k}\phi - \pi({}^0 X_k^{-1} \cdot {}^0 m_{k-1}^i)$ ，我们基于传统或以学习为基础的方法得到的初始光流，引入李代数得到损失函数如下

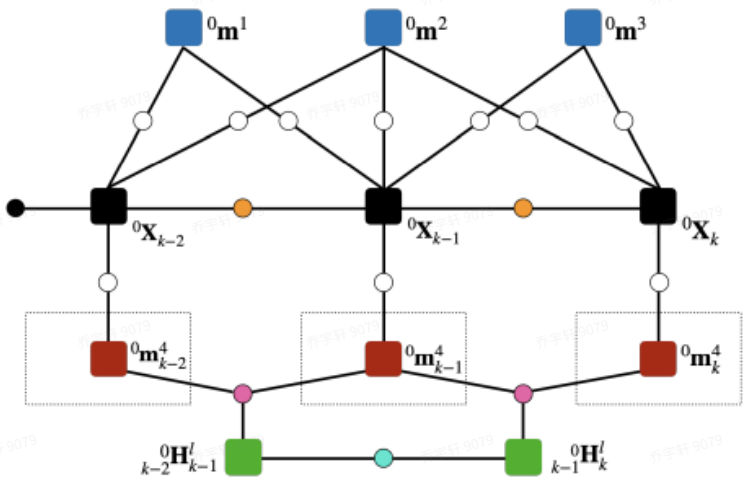
$$ \begin{aligned} \{{}^0\mathbf{x}_k^{*\vee}, {}^k\mathbf{\Phi}_k^*\} = \underset{\{{}^0\mathbf{x}_k^{\vee}, {}^k\mathbf{\Phi}_k\}}{\operatorname{argmin}} \sum_i^{n_b} \Big\{ & \rho_h \left(\mathbf{e}_i^{\top}({}^{I_k}\boldsymbol{\phi}^i)\, \Sigma_\phi^{-1}\, \mathbf{e}_i({}^{I_k}\boldsymbol{\phi}^i) \right) + \\ & \rho_h \left(\mathbf{e}_i^{\top}({}^0\mathbf{x}_k, {}^{I_k}\boldsymbol{\phi}^i)\, \Sigma_p^{-1}\, \mathbf{e}_i({}^0\mathbf{x}_k, {}^{I_k}\boldsymbol{\phi}^i) \right) \Big\}, \end{aligned} \tag{13} $$

其中 $\rho_h(e_i^T({}^{I_k}\phi^i) \cdot \Sigma_\phi^{-1} \cdot e_i({}^{I_k}\phi^i))$ 是关于 $e_i({}^{I_k}\phi^i) = {}^{I_k}\hat{\phi}^i - {}^{I_k}\phi^i$ 的正则化项，其中 ${}^{I_k}\hat{\Phi}^i = \{{}^{I_k}\hat{\phi}^i\}$ 为基于传统或以学习为基础的方法得到的初始光流，Σ_ϕ 为最小二乘的协方差矩阵。类似地，移动物体的误差函数也可以根据光流给出

$$ \begin{aligned} \{{}^0_{k-1}\mathbf{g}_k^{*\vee}, {}^k\mathbf{\Phi}_k^*\} = \underset{\{{}^0_{k-1}\mathbf{g}_k^{\vee}, {}^k\mathbf{\Phi}_k\}}{\operatorname{argmin}} \sum_i^{n_d} \Big\{ & \rho_h \left(\mathbf{e}_i^{\top}({}^{I_k}\boldsymbol{\phi}^i)\, \Sigma_\phi^{-1}\, \mathbf{e}_i({}^{I_k}\boldsymbol{\phi}^i) \right) + \\ & \rho_h \left(\mathbf{e}_i^{\top}({}^0_{k-1}\mathbf{g}_k, {}^{I_k}\boldsymbol{\phi}^i)\, \Sigma_p^{-1}\, \mathbf{e}_i({}^0_{k-1}\mathbf{g}_k, {}^{I_k}\boldsymbol{\phi}^i) \right) \Big\}. \end{aligned} \tag{15} $$

C.图优化

我们将动态SLAM作为一个图优化问题，去细化地估计相机位姿，物体运动以及建立包括静态动态结构的全局一致地图。如下因子图所示



四种测量/观测被考虑到联合优化问题中：3D点测量，视觉里程计测量，动态物体点上的移动变换以及物体的平滑移动观测。图中黑色点表示先验因子，黑色方块表示相机位姿，蓝色方块表示静态点，红色方块表示同一动态点在不同时间下坐标，绿色方块代表动态物体的位姿变换。里程计因子在图中用橘色表示，点测量因子在图中由白色表示，点移动因子在图中用品红色表示，平滑移动因子在图中用青绿色表示。

i)3D点测量模型的误差定义为 $e_{i,k}({}^0X_k, {}^0m_k^i) = {}^0X_k^{-1} \cdot {}^0m_k^i - z_k^i$ ， $z = \{z_k^i\}$ 为所有时间尺度下所有3D点测量的集合，基数为 n_z .

ii)追踪部分通过3D-2D最小化重投影误差提供了一个高质量的相机运动，可以作为里程测量来约束图中的相机位姿，视觉里程计误差定义为 $e_k({}^0X_{k-1}, {}^0X_k) = ({}^0X_{k-1}^{-1} \cdot {}^0X_k)^{-1} \cdot {}_{k-1}^{X_{k-1}}T_k$,其中 $T = \{{}_{k-1}^{X_{k-1}}T_k\}$ 为里程计测量集合， ${}_{k-1}^{X_{k-1}}T \in SE(3)$,集合基数为 n_0 .该误差可以这样考虑 $({}^0X_{k-1}^{-1} \cdot {}^0X_k)^{-1} \cdot {}_{k-1}^{X_{k-1}}T_k = {}^0X_k^{-1} \cdot {}^0X_{k-1} \cdot {}_{k-1}^{X_{k-1}}T_k = {}^0X_k^{-1} \cdot {}^0\hat{X}_k^{-1}$ ， ${}^0\hat{X}_k^{-1} = {}^0X_{k-1} \cdot {}_{k-1}^{X_{k-1}}T_k$ 为根据里程计测量计算得到的位姿，里程计越精确，误差越接近于单位矩阵。

iii)动态物体上点的移动模型误差定义为 $e_{i,j,k}({}^0m_k^i, {}_{k-1}^0H_k^l, {}^0m_{k-1}^i) = {}^0m_k^i - {}_{k-1}^0H_k^l \cdot {}^0m_{k-1}^i$ ，一观察到的运动刚体 l 上的所有点的移动由同一位姿变换 ${}_{k-1}^0H_k^l \in SE(3)$ 给出。

iv)研究表明，在动态SLAM中，结合场景中目标运动的先验知识是非常有价值的。为了避免某些因素(客观存在)导致的物体移动急剧变化，我们在连续运动中引入平滑移动因子 $e_{i,k}({}_{k-2}^0H_{k-1}^l, {}_{k-1}^0H_k^l) = {}_{k-2}^0H_{k-1}^{l\ -1} \cdot {}_{k-1}^0H_k^l$ ，来最小化连续时间下物体的运动变化。

我们仍由李代数的方法给出以上所有项的联合优化，与 ${}^0X_k$ 相似的，我们给出 ${}_{k-1}^0H_k^l$ 的李代数在欧式空间下的对应 ${}_{k-1}^0h_k^{l\vee} \in \mathbb{R}^6$ ， l 为所有移动刚体的标签。记 $\Theta_M = \{{}^0m_k^i\}, \Theta_X = \{{}^0x_k^\vee\}, \Theta_H = \{{}_{k-1}^0h_k^{l\vee}\}$,将所有待优化变量放入 $\Theta = \Theta_M \cup \Theta_X \cup \Theta_H$,将误差最小二乘

$$
\begin{aligned}
\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \Bigg\{ & \sum_{i,k}^{n_z} \rho_h\left(\mathbf{e}_{i,k}^\top({}^0\mathbf{x}_k, {}^0\mathbf{m}_k^i)\, \Sigma_z^{-1}\, \mathbf{e}_{i,k}({}^0\mathbf{x}_k, {}^0\mathbf{m}_k^i)\right) \\
& + \sum_{k}^{n_o} \rho_h\left(\log(\mathbf{e}_k({}^0\mathbf{x}_{k-1}, {}^0\mathbf{x}_k))^\top \Sigma_o^{-1}\, \log(\mathbf{e}_k({}^0\mathbf{x}_{k-1}, {}^0\mathbf{x}_k))\right) \\
& + \sum_{i,l,k}^{n_g} \rho_h\left(\mathbf{e}_{i,l,k}^\top({}^0\mathbf{m}_k^i, {}_{k-1}^0\mathbf{h}_k^l, {}^0\mathbf{m}_{k-1}^i)\, \Sigma_g^{-1}\right. \\
& \qquad \left. \mathbf{e}_{i,l,k}({}^0\mathbf{m}_k^i, {}_{k-1}^0\mathbf{h}_k^l, {}^0\mathbf{m}_{k-1}^i)\right) \\
& + \sum_{l,k}^{n_s} \rho_h\left(\log(\mathbf{e}_{l,k}({}_{k-2}^0\mathbf{h}_{k-1}^l, {}_{k-1}^0\mathbf{h}_k^l))^\top \Sigma_s^{-1}\right. \\
& \qquad \left. \log(\mathbf{e}_{l,k}({}_{k-2}^0\mathbf{h}_{k-1}^l, {}_{k-1}^0\mathbf{h}_k^l))\right) \Bigg\}, \quad (21)
\end{aligned}
$$

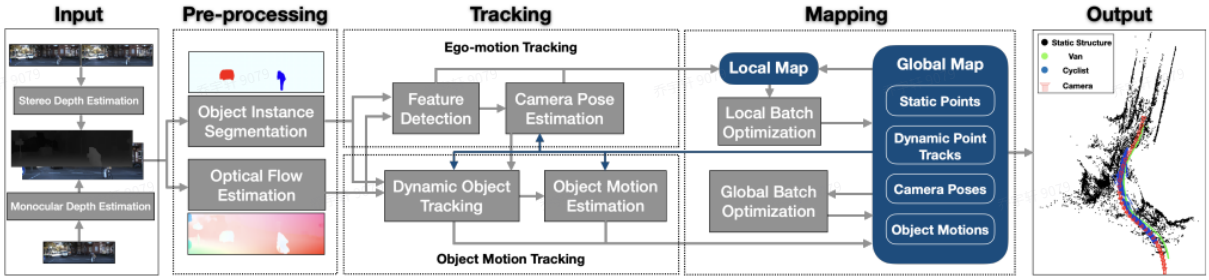
与i),iii)不同的，ii),iv)在最小二乘中引入了log，这是因为ii),iv)问题的误差最小时并不是数值上的误差最小(而是依据接近单位矩阵的程度)，因而我们需要利用对数映射得到一个在最优情况数值上最小的目标函数。

其中 Σ_z 为3D测量噪声的协方差矩阵， Σ_o 为里程计测量的协方差矩阵， Σ_g 为移动噪声的协方差矩阵， n_g 为所有动态物体移动因子的数量， Σ_s 为平滑移动的协方差矩阵， n_s 为所有平滑

移动因子的数量，(21)式可有L-M算法解决。

2. 动态SLAM系统

本部分将展示如何通过第1部分给出的公式将我们VDO-SLAM的思想完善成一个完整SLAM系统



我们的输入必须要求有图像信息与深度信息(主要是利用语义信息与点的3D位置)，虽然原本是为了RGB-D设计，但是对于双目与单目都有办法获得深度信息。

A.预处理

有两个具有挑战性的部分在这个模块需要实现，一是鲁棒的分离静态背景与物体，二是保证长期的动态物体的追踪。我们引入实例级分割(预测对象类标签)与稠密光流估计来解决。

1)物体实例分割：实例级分割被用来分割与识别潜在移动的物体，为我们的分离提供重要先验。另外，分割掩膜提供一个“精确”的物体边界来帮助我们实现物体上点的鲁棒追踪。

2)光流估计：如Dynamic SLAM:TNFS提到的提取特征点办法的局限性，动态物体占比不大的情况下，本就稀疏的特征点匹配更难实现运动物体的鲁棒追踪。我们的办法利用稠密光流通过在语义掩膜中采样可观地增加了物体点的数量。通过给目标掩膜中的每一个点赋予标识符，我们可以进行多物体的一致追踪。区别于稀疏光流，稠密光流可在语义分割失败时恢复物体掩膜。

B.追踪

追踪部分需要两个模块：一是相机移动的追踪，包括特征检测与相机位姿估计；二是物体移动追踪包括动态物体追踪与物体移动估计。

1)特征检测：为了实现快速的相机位姿估计，我们检测一组稀疏的角点特征并利用光流对其进行跟踪。在每一帧中，只有拟合相机运动估计的内点才被保存到地图中，用于跟踪下一帧的对应。如果内点的数量低于某一阈值，系统就会检测并添加新的特征。稀疏特征剔除了分割的物体。

2)相机位姿估计：我们要做的是对相机位姿的初始化(注意在全局建图后，建图结果会重新修正相机位姿)。为了保证鲁棒估计，我们考虑两种估计办法，一是由本文1.B.1)中(13)式的结果得到一位姿估计，二是由基于RANSAC的P3P得到一位姿估计。我们对这两个结果计算重投影误差后的内点个数进行比较。选择产生更多内点的结果进行初始化。

3)动态物体追踪：包括将分割物体进行分类(静态or动态)，再对动态物体在连续帧间进行追踪

我们引入场景流来刻画场景的运动 $f_k^i = {}^0m_{k-1}^i - {}^0m_k^i = {}^0m_{k-1}^i - {}^0X_k \cdot {}^{X_k}m_k^i$ ，f_k^i 为场景流向量。对于静态3D点来说，其场景流大小应当为0，然而噪声与误差会让情况变得复杂。为了鲁棒地解决这个问题，某点的场景流大于某阈值则被认为是动态点，某物体中动态点比例超过某阈值则

被认为是动态物体。系统可以灵活的将静态物体建模为动态(认为是做出了0的运动)，但相反的情况则会降低系统的性能。

实例级别的语义分割仅仅能够提供单张图的分割结果，但是没法实现追踪，因此需要引入稠密光流进行数据关联。使用光流为每个运动物体进行编号，对每个固定物体，由前后两帧对应物体中出现最多的编号标记。对受到噪声，图像边界及遮蔽影响的运动物体，当其上一帧周边的点场景流运动标签为0时，重新对其编号。

4)动态物体移动估计：与先前相机位姿估计相似地，稀疏特征处理相对脆弱。我们采用相似的办法，在掩膜中每三个点进行一次取样，并且追踪。只有与动态物体移动拟合的内点会被保存到地图中，并在下次追踪中被使用。跟踪点数量较少时，会有新的点被采样然后添加。如此得到一个动态物体移动的初始化，在全局建图后会被进一步地修正。后文中还提到了运动速度的估计，总体与Dynamic SLAM:TNFS中相同，添加了对物体中所有点进行平均，使结果更加可靠。

C.建图

在地图模块中，系统会构造并维护一个全局地图。同时，系统会从全局地图中提取出一个基于当前时间步长和前一个时间步长窗口的局部地图。两个地图都是通过批量优化过程更新的。

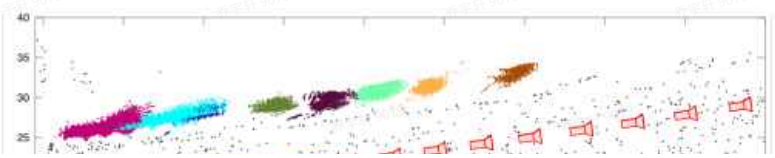
1)局部批量优化：目标为保证精确的相机位姿估计来提供给全局批量优化。局部地图由固定大小的滑动窗口建立信息，局部地图共享一些信息。我们仅仅只局部地优化相机位姿与静态结构，动态结构的优化需要添加一些强约束条件才能带来收益，如匀速运动。(可能动态结构需要一些时间尺度上的连续性) 局部地图中细化并优化变量由因子图来实现。

2)全局批量优化：跟踪模块与局部批量优化输出了相机位姿、物体运动和内点结构。它们都被实时的保留在一全局地图中。当所有输入帧被处理完后，1.C中的因子图将被用来实现全局的优化，当然其中还包括动态结构，优化结果将被输出作为整个系统的结果。

3)从建图到追踪：维护好的全局地图具有大量的历史信息。流程图中的蓝色箭头将内点信息等传递到追踪模块中估计当前帧的相机位姿与物体运动。相机位姿也可作为先验在相机位姿估计与物体移动估计中实现初始化。在语义分割失败的“间接遮挡”下，利用历史的分割掩膜(mask),可以实现跨帧的鲁棒追踪。

总结

在本文中，我们介绍了VDO-SLAM。这是一种新颖的基于动态特征的，基于场景中图像语义信息的，不需要额外的目标位姿或几何信息，就可以实现对动态目标同步定位、建图和跟踪的SLAM系统。我们认为在目标运动估计中之所以能取得高精度结果是由于我们的系统是一个基于特征的系统。特征点(法)仍然是SLAM系统中最容易检测，跟踪和集成的(方法)，不需要前端提供任何关于目标模型的额外信息。文章还提出，如何减少实际情形下动态物体带来的计算复杂度是SLAM中的重要问题之一，下图是VDO-SLAM的一个直观结果图。



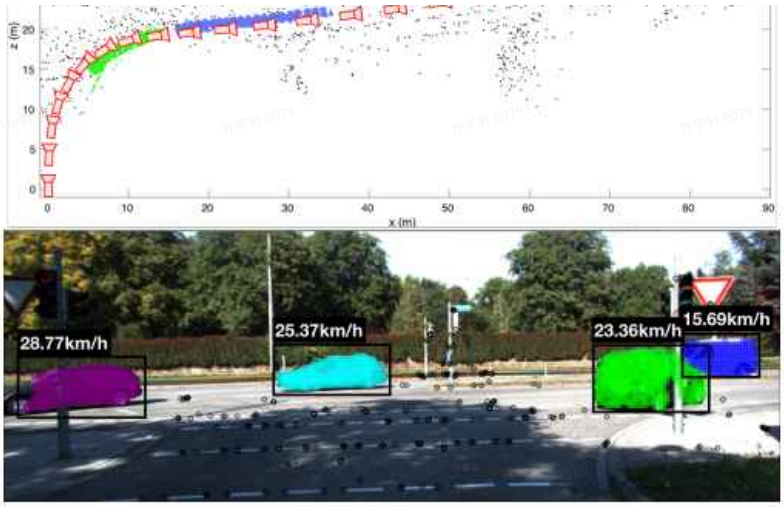


Fig. 1: **Results of our VDO-SLAM system.** (Top) A full map including camera trajectory in red, static background points in black and points on moving objects colour coded by their instance. (Bottom) Detected 3D points on the static background and the objects' body, and the estimated object speed. Black circles represents static points, and each object is shown with a different colour.

存在的问题

- 针对场景流向量设定的数值阈值对于不同情形下是否足够鲁棒，也许可以通过语义分割的静态背景的场景流给出一个参照
- VDO-SLAM是将运动物体作为刚体进行考虑。对于行人等物体来说，他们并不是完全的刚体，VDO-SLAM处理这些物体时表现也许不一定优秀