

主题：未来SLAM讨论

1.未来SLAM应该更加关注机器人对场景的理解，根据场景的动态变化规划自己的路径。

本次讨论中，机器人场景理解是计算机视觉研究一直研究的问题，目前是SLAM与深度学习结合比较多的部分。对于场景的理解是无线追求的，但是得考虑工程应用的必要性和制约性。SLAM现在更多被看成是解决工程问题，比如对场景的理解，这个理论的框架本质上不是属于SLAM这个范畴的，只是整个识别中的一个部分，然后应用到SLAM中。场景的理解还是得根据实际情况的条件而定。

此外还有一个问题就是对于场景理解的定义。这个定义本身比较模糊，讨论过程中由于对于场景理解的定义比较模糊，所以有许多小伙伴讨论起来有点没有准确目标。本人理解机器人场景理解是在运动过程中对周围场景的静态和动态的运动都能跟踪理解然后，做出路径规划和运动控制。现在SLAM中加入一些语义理解例如对于从添加语义标签的语义，来辅助机器人再环境中进行运动。技术流程可能是训练点云，图像，告诉机器人这是冰箱，然后机器人在原来地图上搜索吧

现在深度学习和slam结合的方向可以大致分为两块，松耦合和紧耦合。松耦合是对SLAM框架中一些组成部分进行修改，如前端的位姿估计，后端的建图部分进行修改，深度学习的结果可以作为SLAM框架继续运行的数据。紧耦合在场景理解中的使用深度学习算法构建深度SLAM框架。

2.如果地图能够满足机器人导航的需求，地图的形式不再重要。

讨论中大家觉得地图的形式还是很重要，业界需要有统一的标准来共同打造解决方案。

地图是一种数据格式或者表示方式，应该有统一的标准。当前在智能车上面的高精度地图一般遵守国家上公认的两项协议，我国用opendrive的比较多，不过也可以是地图的形式不一样，但是抽象出来的对外接口有同样的标准。地图表达格式和标准是不是以后会根据传感器类型，或者有更高要求的地图形式表示而改变。

方宇提出拓扑地图这种形式比度量地图在机器人导航里面应该更有优势，不知道大家怎么看？感觉如果是为了机器人导航而服务，可以建立拓扑地图，然后机器人在运动过程中逐渐去判断自己距离当前目标点的距离。王琛提出该是一个技术方向，但是自己计算距离会不会带来更大的计算量。几何度量地图也不一定就不能再减少体积，我觉得都是一些权衡。在拓扑地图里面融入尺度信息。现在我觉得一个工程问题抛出来应该怎么做都是需要和实际应用结合的，看实际应用的限制和需求。

机器人导航我感觉根本不是规划出最优路径局部规划一下，而是局部规划。

3.精度和鲁棒性应该是SLAM肯定要追求的目标，目前机器人还不够智能？

精度在技术上是永远要追求的，鲁棒性是永远要追求的。我的理解是够不够智能也是要看用在什么地方，可能这个场景下已经足够智能了，只不过技术上的目标是希望它像人一样智能。智不智能主要看能不能适应场景。slam的建图过程和人记忆地图的过程有点像。有同学提出现在的gnn因子推理图可能是机器人智能化的趋势。gnn处于萌芽阶段，[介绍GNN因子图的博客](#) [GNN综述文章](#) [GNN综述文章2](#)

王琛：我觉得我需要反问一个问题，理解能力有标准定义吗？我们怎么知道它提高了？

这个问题还是对理解能力定义的标准，目前没有一个明确的标准。王琛建议，那就需要结合具体任务说明了，我同时认为不同种类的任务哪一个更复杂有时候无法区分。但是这些问题比较可能是各自制定的标准，到底哪个大哪个小也没有统一的标准，我认为一旦涉及到更，和提高，必须得有具体的定义和标准。

4.SLAM和深度学习都有哪些结合的点？这些点目前只是前沿探索，还是已经有实际落地应用了？

对于深度学习结合SLAM落地实际应用目前来说还是比较稀少，但是深度学习和SLAM结合的论文和代码还是有不少。例如cubeSLAM是一个深度学习与SLAM结合的比较好的，代码也开源，发表在TOR上的，TMU的CNN-DVO。

如果有传感器数据作为训练数据输入，和深度网络联合训练，然后再运行的时候输入传感器数据，这个思路可行吗？直接端到端进行训练就可以了。还有一个问题，同一个系统中同时运行多个任务的神经网络，三个及以上的时候有没有什么好的方法。不用考虑那么多，拉起直接跑，只要电脑性能好，想跑几个跑几个。有一个比较好的代码环境配置方法，你只用把一个代码搞好，封装成一个镜像。docker并不怎么耗资源，cuda有个docker定制版本的，只能在ubuntu上用，GitHub上能找到。

上周遗留问题讨论

1.舒尔补解

[舒尔补解综述文章](#)

1)什么是舒尔补解

舒尔补是margin的一种方法.舒尔补形式上我认为就是高斯消元，只不过我们手动高斯消元是将 $Ax=b$ 的A转换为上三角(或下三角)，然后一个个元素输出来。只不过舒尔补是先将A通过分块消元，如果将分块看成是 2×2 的矩阵，实际上也就是一个 2×2 矩阵的高斯消元。Sliding window filter with application to planetary landing

2)舒尔补解在高斯消元的时候有什么优势呢？

舒尔补我理解就是在边缘化的时候，为了保留被边缘化的重要信息，从而进行的降低求逆矩阵的维度，提高计算效率，高斯消元也会保存原有矩阵信息，不会改变 $Ax=b$ 的解。

3)舒尔补中分块求逆如果有不可逆的情况时候如何处理呢，实际问题中怎么解决不可逆问题呢？

不可逆，就是 rank deficient，SVD 分解，中间对角阵，element wise 倒数。舒尔补是作用在分块肯定有逆的矩阵上的。

舒尔补这个东西用在slam里，正如前面心亮说的，应该从概率方向来理解。协方差矩阵在去掉一些块的时候，正好对应信息矩阵做舒尔布。或者由多元高斯分布的概率密度做marg也能很清晰的看出来。 $\exp((x-u)^T A (x-u))$ 这个里面A就是信息矩阵，做完marg正好会剩下舒尔补的形式。

2.FEJ问题

因为信息矩阵中同一个状态有不同的展开点，导致零空间发生变化（可观性发生变化。于是引入fej 让相同状态的线性展开点一致 来修正这个变化。这个东西其实可以由非线性最小二乘的法方程来理解： $A dx = -d$ 。这个A其实就是信息矩阵，dx在A的零空间上无论发生什么样的变化 都不会影响这个等式的成立。也就是说在某些方向发生变化，不会影响最小二乘指标。这样很直观的就知道这些维度不可观了，A只可能是因为数值原因奇异。如果是严格的信息矩阵不可能奇异，因为协方差矩阵肯定是可逆的 因此他的逆——信息矩阵 也肯定是可逆的。

A是信息矩阵这个一直不太理解？信息矩阵不就是协方差矩阵的匿吗？有人认为不就是手动设定的吗？对角元非零且对称。JTOmegaJ这个东西就是信息矩阵该有的形式。

游振兴:感觉晨哥刚才那个从方程理解客观性挺有趣的，是不是可以这么理解，A的秩对应可观的维度，其他不可观的对应零空间，所以无法求解优化。其实这个我认为和高斯消元等价。其实是从高斯分布的信息矢量与信息矩阵的表示形式推出来的。 $A=[1,0;0,0]$ ， $x=[0;k]$ ，k随便怎么取，都不影响 $Ax=0$ 。这个方程对应SLAM优化的过程是怎么对应起来呢？就是迭代过程中的法方程，实际 $Hx=b$ 。H的质（md找了半天没找到对的字）=可观状态维数。所以对不可观的维度比如yaw做优化其实是没意义的了是不是可以这么说，相对yaw还是有用的。

不可观的维度没法优化，不管怎么优化都对你的能量没有影响。但从物理意义上来说相对变化还是可观的？一般都是说不可观的维度是全局的，但从物理意义上来说 相对变化还是可观的。

如果信息矩阵精准的话 直接看零空间对应它的哪些维 然后咱们知道哪些维是哪些状态 就知道哪些状态不可观。我们在初始化的时候其实是固定了这些不可观的初值，比如yaw=0,全局位置等于0。感觉是不是可以从这个角度看数学上有什么变化，如果是固定不可观（零空间维度上）的值，对最后的cost function收敛理论上是没有影响的。

王琛：但是有时候看不出是具体哪个维度不可观，比如尺度（随便举个例子），有时候可能是参数化的问题，如果某种参数化方法能把这些维度隔离开，那就又不一样。

笑晨：但是有时候看不出是具体哪个维度不可观，比如尺度（随便举个例子），有时候可能是参数化的问题，如果某种参数化方法能把这些维度隔离开，那就又不一样。这个东西反正我没想透，我经常这样想：比如在某一次SW里面，它已经有先验的yaw或者其他东西，如果零空间正常，这些东西应该就不会有不正常的变化（它们的变化只受相对约束的影响），但是如果人为加一个变化，在法方程上也反应不出来。 [EKF论文](#)