自动驾驶感知算法

入门 进阶 实战

1 Part 为什么设计这个系列课？

1、这个方向有发展潜力

热门方向

前景

自动驾驶、VR、游戏等

钱景

应用广

薪资都挺高

缺口大

学习难度大，人才少

2、学习资源少，难度高

招聘要求

SLAM算法工程师

职责描述：

参与研发基于多相机和IMU的实时SLAM系统，并在VR眼镜上的部署和调优。

任职要求：

1.硕士及以上学历,自动化、计算机、电子工程、通信工程等相关专业；1年以上机器视觉研发经验，有VR\AR\MR平台SLAM研发经验优先；

2.熟练使用相关的C++库：OpenCV，Ceres，OpenGV，Eigen等；熟练使用STL容器；

3.概率论、矩阵论和计算机视觉原理等基础扎实；

4.良好的英文读写能力；

5.熟悉视觉SLAM系统的前端Tracking，后端优化，Loop Closure等模块；

6.熟悉相机和IMU的内外参标定；

7.熟悉Pinhole，KB8，MEI，Radial-Tangential等相机模型；

8.熟悉VIO或者SLAM系统，参与过基于VINS-Fusion，OpenVins，MSCKF，ORB-SLAM等的研发工作；参与过实际落地项目者优先；

9.有曝光调节，SEE加速，CUDA开发经验者优先；

▲ 赞同 13 ▼ 10 条评论 分享 收藏 喜欢 ··· 收起

2 Part 这个系列课有什么？

系列包含七门课程

课程	内容介绍
C++编程实战	语言基础，主要的开发语言
Python编程实战	语言基础（脚本语言），计算机视觉需要
PCL编程实战	点云处理
ROS/ROS2编程实战	算法需利用ROS的消息订阅发布机制，如cartographer_ros
计算机视觉与OpenCV算法实战	融合视觉信息，视觉SLAM中特征提取都需要
SLAM算法实战	SLAM数学基础，SLAM基础，激光/视觉SLAM算法讲解
多传感器融合实战	传感器介绍，传感器标定，传感器数据融合

3 Part 系列课为什么这么设计？

先看一个高赞回答

258 人赞同了该回答

个人：双非硕，无实习，有论文比赛专利（与slam无关的）。学了一年多的slam，实验室之前没人做过slam，自己第一个学这块的，报了挺多网课，学到了挺多，推荐。

面试：之前搜slam就业都是各种劝退，我还是双非的，一开始做好了找不到slam找机器视觉的准备，没想到最后结果还挺满意。先找了几个小厂试水，有拿到2个20W左右的，后面有点信心就投了有点名气。最后拿了几个不错的offer。面试的时候有试过被问到头晕，不过那几个难的面试反而是学到最多的，复盘很重要，面试被问到不懂反而是学得最多的时候，所以多面试几家吧。

面试都是问算法题，C++，项目。

算法题：一般的是剑指offer的难度，大厂或独角兽就不够，好几家都是挂在笔试算法题上，参照别人的意见，刷够200题吧。

C++：这个我学得也一般，有面试是挂在这里。基本是问那几个，虚函数作用，多线程锁智能指针，const，static关键字作用。

Slam知识：基本是十四讲的东西，对极几何，李代数作用。Pnp求解需要几个点。

有个项目跟vins比较相关，被问了挺多vins的东西，基本是vins论文的东西，在线标定，初始化流程，预积分作用，回环跟orbslam2区别。

赞同 258　127 条评论　分享　收藏　喜欢　···　收起

先看一个高赞回答

三就是个人基础了，如果数学（矩阵论，概率论等等）和编程（c++）能力强可以节省大量时间，否则要花大量时间学这些东西

四.除了上面基本数学和编程之外，slam本身涉及到的理论很多，ros，前端里面的:初始化，在线标定，特征提取，特征匹配，光流追踪，pnp，对极几何，激光常用的ICP,局部BA，后端里面的因子图优化，BA, 各种滤波算法（KF,EKF,ESKF,IEKF,INEKF.....）等等，这些只是最基本的模块，其中后端理论是slam理论最难的部分。具体到slam算法又有很多框架:纯视觉（svo，dso，orb系列等），视觉惯性（vins系列，msckf系列，Rovio等），激光之类的（loam，fastlio等），还有很多多传感器融合算法（r3live，lio sam等），这些不说全部看完，起码要完整学完几个框架吧，在看代码过程中又会发现有很多库要会用（eigen,opencv,pcl,pangolin,ceres,g2o），要学的东西太多了，不花大量时间连门都进不去，更何况精通一些框架了

▲ 赞同 42 ▼ 9 条评论 分享 收藏 喜欢 ··· 收起

再看招聘要求

数学基础

计算机视觉算法

职位描述

C++ 算法设计 视觉图像算法 运动控制算法

工作职责:

-基于非线性优化理论对建图算法进行优化，提升建图的速度与精度和鲁棒性

-对定位算法进行优化，通过策略剔除错误观测数据，提高定位的精度和鲁棒性

-根据产品需求，对建图和定位算法进行工程化实现，达到预期效果

-定位数据上下游输入输出数据流和条件进行判断，对异常情况进行诊断和处理

-配合实车测试联调相关算法和功能，跟进解决相关问题等

职位要求:

-本科以上学历，扎实数学功底（矩阵运算、微积分、优化算法），熟悉经典的多视角几何基础理论，熟悉计算机视觉中的常用的非线性优化理论

-熟悉Linux，熟练掌握C++编程，熟悉ceressolver等优化库的使用

-熟悉机器人 / 自动驾驶相关多传感器融合定位算法，包括基于视觉与IMU、轮速等的Kalman相关定位滤波算法

-熟悉经典的SFM/SLAM方法，熟悉ORB-SLAM，VINS等一种或多种经典算法，熟悉经典的vio优化算法

-熟悉一种或多种特征提取算法如SIFT、ORB、SuperPoint等，熟悉众包场景下的地图拼接、融合算法优先

-了解相关硬件特性（相机、IMU、Encoder etc.），熟悉一种或者多种传感器内 / 外参标定算法优先

-具备较强的责任心、良好的沟通能力和团队合作意识，善于思考和分析解决问题

2

百度 · SLAM算法工程师

Linux使用

C++编程

多传感器融合

SLAM算法

ORB-SLAM

多视图几何

非线性优化

卡尔曼滤波

传感器

多传感器标定

4 Part 课程优势是什么？

1 零基础入门

2 内容全面详细

3 一对一答疑

4 课程持续更新，永久有效

5 理论与实战结合

6 提供讲义与代码

7 面向工程与求职

8 讲师经验丰富

肖老师，中国科技大学硕士，自动驾驶算法工程师，《学习OpenCV4：基于Python的算法实战》和《深度学习计算机视觉实战》两本图书作者，从事自动驾驶感知融合与SLAM建图算法开发，分享深度学习、计算机视觉、OpenCV、自动驾驶、SLAM、C++/Python开发等方向的内容。

SO(3)李代数so(3)详细推导

任意的旋转矩阵 $\mathbf{R}$，有 $\mathbf{R}\mathbf{R}^T=\mathbf{I}$

$\mathbf{R}$ 在随时间不断变换，引入时间t 就得到 $\mathbf{R}(t)$，上式也变成

$$\mathbf{R}(t)\mathbf{R}(t)^T=\mathbf{I}$$

等式两边一起对时间求导可得：

$$\underbrace{\dot{\mathbf{R}}(t)}_{\text{导数}}\mathbf{R}(t)^T+\mathbf{R}(t)\dot{\mathbf{R}}(t)^T=0$$

反对称矩阵

$$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T=-\mathbf{R}(t)\dot{\mathbf{R}}(t)^T=-\left(\dot{\mathbf{R}}(t)\mathbf{R}(t)^T\right)^T$$

$$\underbrace{\boldsymbol{a}\times\boldsymbol{b}}_{\text{向量表示}}=\underbrace{\begin{bmatrix}0&-a_3&a_2\\a_3&0&-a_1\\-a_2&a_1&0\end{bmatrix}}_{\text{反对称矩阵表示 }\mathbf{A}}\boldsymbol{b}\overset{\text{def}}{=}\underbrace{\boldsymbol{a}^\wedge}_{\text{反对称表示}}\boldsymbol{b}$$

两种写法表示：$\mathbf{a}^\wedge=\mathbf{A}$ $\quad\mathbf{A}^\vee=\mathbf{a}$

$\dot{\mathbf{R}}(t)\mathbf{R}(t)^T$ 对应的向量为 $\phi(t)\in R^3$，就有

$$\underbrace{\dot{\mathbf{R}}(t)\mathbf{R}(t)^T}_{\text{对应上面的A}}=\underbrace{\phi(t)^\wedge}_{\text{对应上面的a}}$$

上式两边同时右乘 $\mathbf{R}(t)$ 则有 $\dot{\mathbf{R}}(t)=\phi(t)^\wedge\mathbf{R}(t)$

每对旋转矩阵求一次导数，只需要左乘一个 $\phi(t)^\wedge$ 就可以了。

泰勒展开式：

$$f(x)=f(x_0)+\frac{f'(x_0)}{1!}(x-x_0)+\frac{f''(x_0)}{2!}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+R_n$$

$$\mathbf{R}(t)\approx\mathbf{R}(t_0)+\dot{\mathbf{R}}(t_0)(t-t_0)$$

设t0=0时 $\mathbf{R}(0)=\mathbf{I}$ $\Rightarrow$ $\mathbf{R}(t)\approx\mathbf{I}+\phi(t_0)^\wedge\cdot(t)$

同时，在t0 附近，设 ϕ 也是一个常数，即 $\phi(t_0)=\phi_0$ $\Rightarrow$ $\dot{\mathbf{R}}(t)=\phi_0^\wedge\mathbf{R}(t)$

$\dot{\mathbf{R}}(t)=\phi_0^\wedge\mathbf{R}(t)$ 是一个关于 $\mathbf{R}$ 的微分方程，而且知道初始值 $\mathbf{R}(0)=\mathbf{I}$，他的解为

$$\mathbf{R}(t)=\exp(\phi_0^\wedge t)$$

由于我们之前做了一些假设，所以这个式子只在t **附近有效。**

当某个时刻的 $\mathbf{R}$ 已知时，存在一个向量 ϕ，二者满足这个矩阵指数关系。

而这个 ϕ，就是对应到SO(3) 上的李代数so(3)

$$\dot{\mathbf{R}}(t)=\phi_0^\wedge\mathbf{R}(t)\qquad\mathbf{R}(t)=\exp(\phi_0^\wedge t)\qquad P(x)=-\phi_0^\wedge$$

一阶线性齐次微分方程 $\frac{dy}{dx}+P(x)y=0$ 的通解 $y=Ce^{-\int P(x)dx}$ $\qquad -\int P(x)dx=\phi_0^\wedge t$

感谢您的聆听