# 需求分析

宿舍楼到学校西门距离d=2km

小车送餐一次，一来一回，距离为2d=4km

小车行驶速度v=12km/h

要在中午三小时内送完1000分外卖，T=3h，N=1000

小车跑一个来回所需时间t=4km/（12km/h）=1/3h

算上等待时间小车一个来回的时间t=2/3h

3h内一辆小车能跑的来回数3h/（2/3h）大约为5次

每辆小车能装20份外卖

一辆小车一中午能运送20×5=100份外卖

需要1000/100=10辆小车

# 车体

车体尺寸:网上查到，一般的外卖餐盒是750ml容积的。考虑到有的外卖餐盒会较大，并且设计要留有余地，现假设每个餐盒的容积为1000ml。在淘宝上查到1000ml的标准外卖餐盒的尺寸如下。



因为不能让小车重心太高，所以要尽量让小车面积大一些。经测量，半张课桌的面积，如果餐盒的长边沿课桌的长边摆放，餐盒的短边沿课桌的短边拜访，大概可以摆下4×4=16个外卖餐盒。因为要留有余地，而且一辆小车只用装运20份外卖，所以可以把外卖分两层，一层装10份外卖。按照4×3的方式进行摆放，12-10=2，多余的2个空间留作备用。于是小车上货箱的尺寸大体如下：

4×17=68cm，货箱的长边设计为70cm

3×12=36cm，货箱的短边设计为40cm

6×2=12cm，货箱的高度设计为20cm





淘宝链接[外卖箱](https://item.taobao.com/item.htm?spm=a230r.1.14.61.55333a7dFutRHL&id=571820070223&ns=1&abbucket=14" \l "detail)

车轮尺寸：14×2.5

14指的是适应轮圈口径也就是车圈14英寸，2.5指的是轮胎粗细2.5英寸。

14英寸是35.56cm。



车体重量

Pu材料的密度：1.25g/cm3

货箱的重量：1.25×70×40×20=70000g=70kg



现假设整车重量为100kg

电池容量

电动车行驶时电流大小为10A左右，小车要运行3小时，所以容量为30AH。

给电动机供电的电压一般为48V，所以能量为48×30=1440WH。考虑到转化效率为略小于90%，所以所需电池能量为1600WH。

电机

查阅资料得知，电动车电机的输出功率，与车速没有任何关系，车速与电机额定转速相关。

因此需要的电机额定转速应大于333cm/s÷17.78cm≈19rad/s

# 控制器

电调控制电机

飞控直接控制舵机

# AI驾驶

硬件平台：TX2

功能：道路检测、交通标志识别、障碍物检测、行人检测

Yolo

# SLAM



* 传感器信息读取：在视觉SLAM中主要为相机图像信息的读取和预处理。
* 视觉里程计：估算相邻图像间相机的运动，以及局部地图的样子。
* 后端优化：后端接受不同时刻视觉里程计测量的相机位姿，以及回环检测的信息，对它们进行优化，得到全局一致的轨迹和地图。
* 回环检测：判断机器人是否到达过先前的位置。如果检测到回环，它会把信息提供给后端进行处理。
* 建图：根据估计的轨迹，建立与任务要求对应的地图。

经典的视觉SLAM框架是过去十几年的研究成果。这个框架本身及其所包含的算法已经基本定型，并且已经在许多视觉程序库和机器人程序库中提供。如果把工作环境限定在静态、刚体，光照变化不明显、没有人为干扰的场景，那么这个SLAM系统是相当成熟的了。我们的主要挑战是为了降低成本，将不使用激光雷达，和尽量使用单目摄像机。

由于单目相机拍摄的图像只是三维空间的二维投影，所以如果想恢复三位结构，必须改变相机的视角。因此在单目SLAM中，我们必须移动相机，才能估计它的运动。但是即使我们知道了物体远近，它们仍然只是一个相对的值，比如看电影时虽然知道电影场景中哪些物体比另一些大，但无法知道电影中那些高楼大厦是不是模型。单目SLAM无法仅凭图像确定这个真实尺度，称为尺度不确定性。因此，单目SLAM要求我们提供一些先验知识，比如事先知道一些参考物的真实尺度。

建图。对于小车来说，运动环境是校园的地面，近似是一个平面，就只需要一个二维的地图，标记哪里可以通过，哪里存在障碍物，就够了。具体来说，我们需要度量类地图（Metric Map）里的稠密地图（Dense Map）。稠密地图通常按照某种分辨率，由许多个小块组成。对于二维度量地图是许多个小格子（Grid）。一个小块含有占据、空闲、未知三种状态，以表达该格内是否有物体。当查询某个空间位置时，地图能够给出该位置是否可以通过的信息。

Table 1 常用开源SLAM方案

|  |  |  |
| --- | --- | --- |
| 方案名称 | 传感器形式 | 地址 |
| MonoSLAM | 单目 | <https://github.com/hanmekim/SceneLib2> |
| PTAM | 单目 | <http://www.robots.ox.ac.uk/~gk/PTAM/> |
| ORB-SLAM | 单目为主 | <http://webdiis.unizar.es/~raulmur/orbslam/> |
| LSD-SLAM | 单目为主 | <http://vision.in.tum.de/research/vslam/lsdslam> |
| SVO | 单目 | <https://github.com/uzh-rpg/rpg_svo> |
| DTAM | RGB-D | <https://github.com/anuranbaka/OpenDTAM> |
| DVO | RGB-D | <https://github.com/tum-vision/dvo_slam> |
| DSO | 单目 | <https://github.com/JakobEngel/dso> |
| RTAB-MAP | 双目/RGB-D | <https://github.com/introlab/rtabmap> |
| RGBD-SLAM-V2 | RGB-D | <https://github.com/felixendres/rgbdslam_v2> |
| Elastic Fusion | RGB-D | <https://github.com/mp3guy/ElasticFusion> |
| Hector SLAM | 激光 | <http://wiki.ros.org/hector_slam> |
| GMapping | 激光 | <http://wiki.ros.org/gmapping> |
| OKVIS | 多目+IMU | <https://github.com/ethz-asl/okvis> |
| ROVIO | 单目+IMU | <https://github.com/ethz-asl/rovio> |

Jetson TX2真正解决了 SLAM的计算需求，机器视觉是一个延迟敏感的问题，最好的解决方式就是在本地进行计算处理。而 Jetson TX2 拥有强大的计算能力，在处理 720p 彩色图像时的每秒帧数可达 100 帧以上。

ORB-SLAM 是西班牙 Zaragoza 大学的 Raúl Mur-Arta 编写的视觉 SLAM 系统。 它是一个完整的 SLAM 系统，包括视觉里程计、跟踪、回环检测，是一种完全基于稀疏特征点的单目 SLAM 系统，同时还有单目、双目、RGBD 相机的接口。其核心是使用 ORB (Orinted FAST and BRIEF) 作为整个视觉 SLAM 中的核心特征。

ORB-SLAM 基本延续了 PTAM 的算法框架,但对框架中的大部分组件都做了改进, 归纳起来主要有 4 点:

1. ORB-SLAM 选用了 ORB 特征, 基于 ORB 描述量的特征匹配和重定位, 都比 PTAM 具有更好的视角不变性。此外, 新增三维点的特征匹配效率更高, 因此能更及时地扩展场景。扩展场景及时与否决定了后续帧是否能稳定跟踪。
2. ORBSLAM 加入了循环回路的检测和闭合机制, 以消除误差累积。系统采用与重定位相同的方法来检测回路(匹配回路两侧关键帧上的公共点), 通过方位图 (Pose Graph) 优化来闭合回路。
3. PTAM 需要用户指定 2 帧来初始化系统, 2 帧间既要有足够的公共点, 又要有足够的平移量. 平移运动为这些公共点提供视差 (Parallax) , 只有足够的视差才能三角化出精确的三维位置。ORB-SLAM 通过检测视差来自动选择初始化的 2 帧。
4. PTAM 扩展场景时也要求新加入的关键帧提供足够的视差, 导致场景往往难以扩展. ORB-SLAM 采用一种更鲁棒的关键帧和三维点的选择机制——先用宽松的判断条件尽可能及时地加入新的关键帧和三维点, 以保证后续帧的鲁棒跟踪; 再用严格的判断条件删除冗余的关键帧和不稳定的三维点，以保证 BA 的效率和精度。

我们选择ORB-SLAM作为我们的方案，有如下考虑：

* 一个代码构造优秀的视觉 SLAM 系统，非常适合移植到实际项目。
* 采用 g2o 作为后端优化工具，能有效地减少对特征点位置和自身位姿的估计误差。
* 采用 DBOW 减少了寻找特征的计算量，同时回环匹配和重定位效果较好。
* 重定位：比如当机器人遇到一些意外情况之后，它的数据流突然被打断了，在 ORB-SLAM 算法下，可以在短时间内重新把机器人在地图中定位。
* 使用了类似「适者生存」的方案来进行关键帧的删选，提高系统追踪的鲁棒性和系统的可持续运行。
* 提供最著名的公共数据集（ KITTI 和 TUM 数据集）的详尽实验结果，以显示其性能。可以使用开源代码，并且还支持使用 ROS。
* 网上关于ORB-SLAM的资源远比其它SLAM方案要丰富，而且已经有在TX2上实现该方案的先例。
* TX2的GPU运算能力足够，足以支持如此大的计算量。

我们预计将使用单目或双目摄像头+IMU的方式来实现该解决方案，用车轮编码器来为单目摄像头提供绝对尺度信息。

# 传感器

超声波：车体四周，前后左右各放一个

摄像头：放在车顶，三个拼接成360°视角

# 决策

强化学习

# 数据融合