西电机器人夏令营

SUMMER CAMP 2018

七组技术报告

进度安排表

项目	工期	开始时间	结束时间
1、规划、方案讨论	1天	7.26	7.27
2、方案初步设计			
机械：确定机械方案、验证可行性、方案设计以及加工	7	7.28	8.3
嵌入式：确定小车行驶方案、传感器调试	7	7.28	8.3
算法：实现识别魔方、判断魔方状态	7	7.28	8.3
3、方案的讨论、迭代、测试与调试、优化			
机械：改进机构、结构优化	7	8.3	8.10
嵌入式：调试小车、提高鲁棒性	7	8.3	8.10
算法：与嵌入式进行联调，提高程序稳定性	7	8.3	8.10

目录

1 机械部分

1.1 设计动机

因为场地较为平整，所以采用没有悬挂系统的方案，我们采用的是双层底盘的结构。第一层用于放置魔方，主控板，树莓派；第二层用于放置机械臂。

1.2 设计需求

1.比赛的功能需求：稳定的底盘，实现魔方的夹取
2.各需求的优先级：稳定的底盘，实现魔方的夹取
3.大概的设计时间：一晚上

1.3 设计方案

1.第一层底盘：通过电机固定架固定电机，刚性连接，保证绝对的电机位置，以此来保证四个轮子的绝对平行。为了方便布线，在底盘的对应位置开设布线孔。同时也留出位置给第二层的固定柱。

2.第二层底盘：为了同时兼顾实用性与制作的简单性，我们不采用夹取的方式获取魔方，而采用钩子的方式将魔方从柱子上钩下，这样既方便设计又简单实用。同时钩子的伸缩，采用曲杆联动的方式，实用一个舵机控制伸缩即可。

1.4 方案的优点与不足

1.方案的优点：双层底盘的设计，使得底盘较低，在第一层与第二层之间有放置魔方的空间，并且也方便电控组的同学插线布线。

2.方案的不足 ：第一层底盘底盘的重量有些重，长期测试有可能会导致轮子的位置变化、联轴器的变形；魔方在钩下之后，是否能精准的掉落在第一层板之上还有待测试；第一层与第二层之间的柱子稳定性还有问题。第二层底盘钩子的行程不够，需要多次改变第二层底板的长度；曲杆联动的曲杆设计要求极为严格，必须事前计算好所有的尺寸再开始加工；

舵机的中心必须和曲杆的中心在一条直线上，才可以最大限度的进行曲杆的伸缩控制。

2 嵌入式部分

2.1 整体方案

通过遥控器控制小车按预定路线走出迷宫，实现夹取魔方的功能，记录遥控器的数据，比赛时读取相应数据，“录播”完成比赛目标，可以避免传感器数据不准确的问题

2.2 运动学解算方法

1.测速方式

基于编码器原理，利用外部中断测速，为满足pid调速，测速频率应高于控制频率所以选择通过每一个返回脉冲时间得到脉冲数每秒，通过读取另一相电平高低判断正负，并舍弃坏值

2.基于麦克纳姆轮的运动解算

将底盘的运动用三个独立变量来描述：X轴平动、Y轴平动、yaw 轴自转，根据底盘的运动状态解算出四个轮子的速度

3.底盘PID控制

为实现速度的闭环控制，选择在100Hz控制主频下将解算目标速度与实测速度的偏差的比例(P)、积分(I)、微分(D)通过线性组合构成控制量，对速度进行控制。

2.3 机械臂与控制方案

机械臂是一个通过曲轴连杆的机构控制，利用一个MG955的舵机驱动曲轴，控制pwm占空比改变角度，伸出机械臂，反向转动则收起机械臂，实现“挖”下魔方。

底盘通过四个麦克纳姆轮驱动，四个电机驱动麦轮，通过麦轮解算得到每个轮子单独速度，并且将目标速度与实测速度利用位置式pid校准。

2.4 功能模块说明

1.mpu6500陀螺仪

使用mpu6500，SPI通信获取原始三轴加速度及角速度，再通过陀螺仪姿态解算利用四元数等得到当前yaw,pitch,roll三轴姿态角

2.超声波测距

四个方向超声波测距，在15ms定时器中断里，轮流给Trig引脚发送10us高电平，并通过定时

器捕获模式，改变中断上，下升沿中断触发模式，读取时间差再转换为距离

3.DJI遥控器

串口通信，并且对数据进行解包，读取遥控器信息

2.5 难点与不足

1.为实现pid的控制效果，测速也在不断优化，最初想法外部中断计数，定时器中断测速，为提高测速频率，选择读取10次脉冲记录时间进行一次测速，最终选择每个脉冲测一次速，并过滤坏值。并且判断速度正负方法由读取输出pwm定时器寄存器状态改为读取编码器另一相电平状态判断速度正负。

第3章 算法部分

3.1 开发环境介绍

1..在 ubuntu16.04 下 opencv 库 + apriltag 库 进行程序魔改在树莓派 3B ubuntu mate 下运行

2.遇到的坑：安装 opencv3.4.1 版本时，发现新增了内存保护一类的宏 有很大概率导致播不了视频（解决：卸载 注释掉 啥 avi.cpp 里的某一行 从新编译安装）

3.2 整体技术方案概述

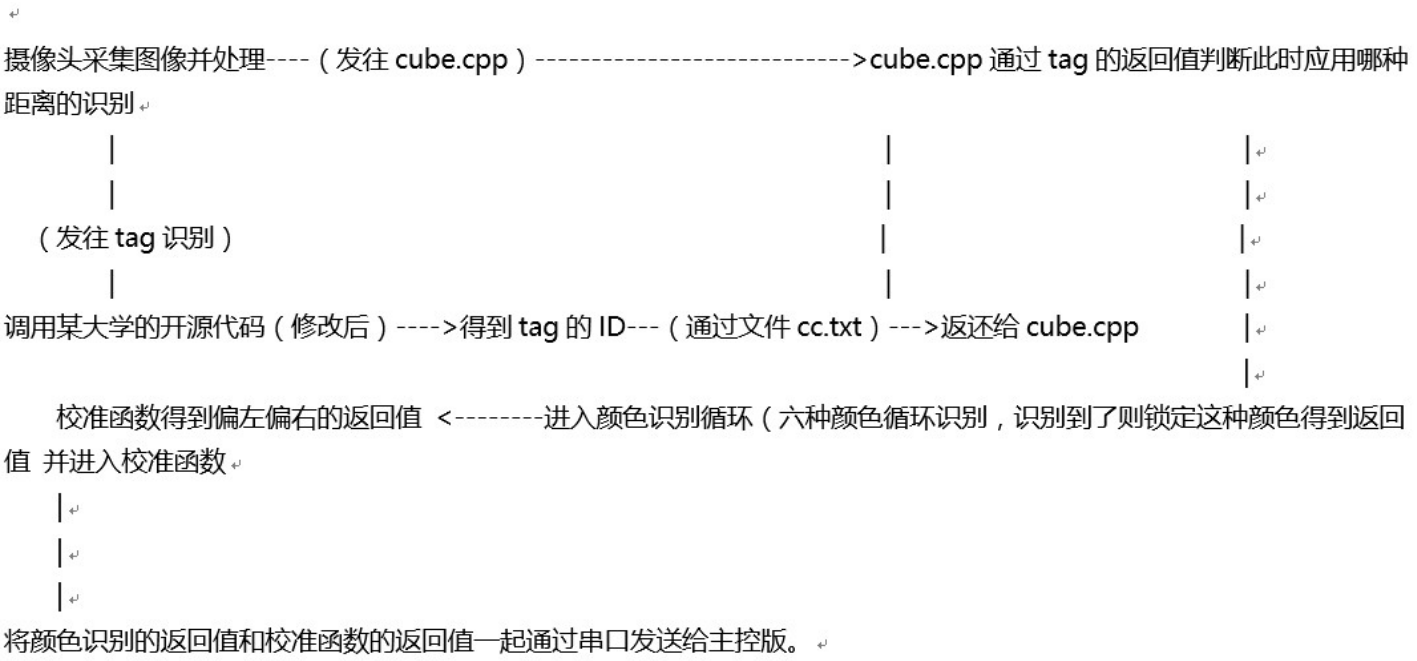
1.opencv 库函数的调用 2.apriltag 源码的改造

3.基于魔方外边沿轮廓进行放大以实现远距离识别

4.对模仿的颜色识别用 HSV 和 Lab 色彩空间按需使用 基本消除了一开始出现的误识别情况

5.用文件及图片的方法解决了两个不同程序之间的图像和信息传输 （虽然我是想用本地 socket。。。。但是当时太困了）

3.3 算法整体框架设计



3.4 算法功能模块说明

1.采集图像：收集用于识别的基本元素 并通过处理使得所需要的特征更明显的暴露出来 2.tag 识别：识别场地中的 tagID，来进行辅助导航和识别模式的切换 （tag 识别可以与魔方识别同时进行

3.魔方识别：舍弃了原来的轮廓识别 直接对颜色识别 在识别某一中颜色时 若返回的旋转举行个数==9 则可以判断该魔方必定还原。优势：避免了原有的光强干扰 缺点：颜色参数过多需要细调。

4.校准 ： 在夹取魔方时，通过中心块的 center 与最外轮廓（视野）的 center 水平坐标的差值正负判断此时是否正对魔方

。

3.5 测试结果

1.图形采集：在高亮的情况下摄像头曝光会很高导致颜色识别出现一些误差2.tag识别：表现良好基本无失误。3.颜色识别： 红与白识别效果较差 参数受摄像头曝光影响4.校准：表现良好。

3.6 可优化方案

1.目前运行速度还可以更快:拆分成几个程序 用socket进行程序间通信2.引入深度学习可以将导航完全用视觉完成3.甚至通过遥控器训练 进行最短路径的建模 让小车自己分析最短路径（类似菲斯卡尔智能小车比赛）

4 夏令营感想、总结

作为夏令营的 staff 准备和设计方案的时间不足，导致后期任务时间紧促，许多功能没有足够的时间实现，影响了整个进度……