

工程实践与科技创新 III -F

结题报告

项 目 名 称 : 基于单眼视觉的小车自动巡线和倒车入库系统

姓 名 : 刘启明

学 号 : 516021910585

联 系 电 话 : 13262921489

电 子 信 箱 : liuqiming666@sjtu.edu.cn

合 作 人 姓 名 : 赵寅杰、严威豪

2019 年 6 月

1 研究内容综述

本课程以树莓派和后驱四轮小车为平台，基于 RGB 摄像头取得图像，对图像进行处理和信息提取，最后根据任务要求对小车进行相应控制。我们需要完成两个任务：任务一（后称巡线任务）要求小车开启前置摄像头，观察前方地面上的黑色轨迹线，并沿着黑色轨迹线前进直到终点；任务二（后称入库任务）要求小车开启后置摄像头，观察后方车库，并根据车库顺序依次倒车入库。

对比开题报告中确定的研究目标，我将在本节中简要介绍各目标和性能指标的完成情况，并对我们小组的项目效果作简要概括。

对于巡线任务，我们小组的预期目标为：无人干预的情况下小车顺利到达巡线终点；在巡线过程中不偏离中心线，黑色中心线始终在两轮之间；小车在 60 秒之内走完全程。经过我们多次重复测试，我们的程序都能够让小车顺利走完全程，走一圈所用时间大约为 20 秒，这远远高于预先设定的指标。另外我们小车在行进的过程中能够及时调整与车道中心线的偏移，调整过程平滑，前轮无明显抖动，小车行进过程中，黑色中心线始终在两轮之间。以上事实说明，我们的巡线任务能够在保证巡线准确性的前提下，尽量使小车以更快的速度移动。具体信息见结题视频，巡线的实现方法将在第二部分中详细介绍。

对于入库任务，我们小组的预期目标为：小车能够顺利倒车进入任意指定车库；小车倒车入库的过程中不触碰边线，且入库后姿态端正。经过多次重复测试，我们能够使小车在不完全正对车库的情况下自行调整入库，且能够依次进入各并排的车位。在进入车位的过程中，小车能够自动调整以避免触碰边线，进入车库之后车体较为端正，与边线的夹角不超过 15° 。具体信息见结题视频，入库的实现方法将会在第二部分中详细介绍。

综合来看，我们小组较好地完成了在开题报告中所预设的目标，有些指标甚至超过了期望值。对于巡线和入库两个任务，我们不仅完成了基本要求，还完成了所有的附加技术要求，最终的实现效果令人满意。我将在第二部分的研究方案中重点介绍两个任务的具体实现方法和一些技术细节。

2 研究方案

在本节中，我将详细介绍我们巡线和入库任务的技术实现途径。出于行文条理考虑，我不在本节中进行实现方案的讨论（如对曾经尝试方法的介绍），仅仅是对我们的最优方案进行全面的介绍，以逻辑顺序为序。当然，项目中我们也遇到了很多困难、尝试了多种方法，相关讨论将会在 4.1 节中详细介绍。

2.1 巡线任务的实现

2.1.1 图像处理部分

本任务中，白色背景地面上粘贴着由黑色胶带围成的闭合曲线。我们需要控制小车沿着

黑色轨迹线行进。黑色轨迹线的形状如图 1 所示。

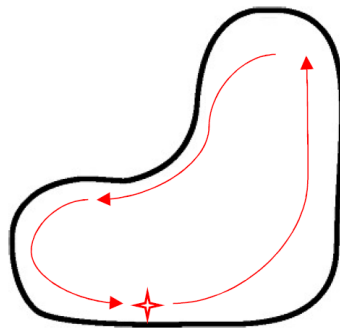


图 1 巡线任务中的轨迹线

由于轨道线颜色与背景颜色反差很大，我们想到先对图像进行灰度化、二值化，再在二值化图像中进行特征提取工作。另外由于摄像头视角不利于后期调试和特征提取，我们在处理之前先对图像进行了透视校正。

首先介绍透视校正的工作。透视校正是将一个侧视视角图像转化为一个俯视视角图像的校正方法，采用这种校正可以滤除大量噪声，并去掉被观察物体的透视畸变（即“近大远小”现象）。在图 2 中，摄像头获得的原始图像如左侧图所示。选定我们的目标区域之后，对图像相应的点进行拉伸与压缩，即可得到透视校正之后的图像，如右侧图所示。观察右侧图，可以发现透明文件夹近似为一个矩形，图像被校正为一个近似的俯视图。

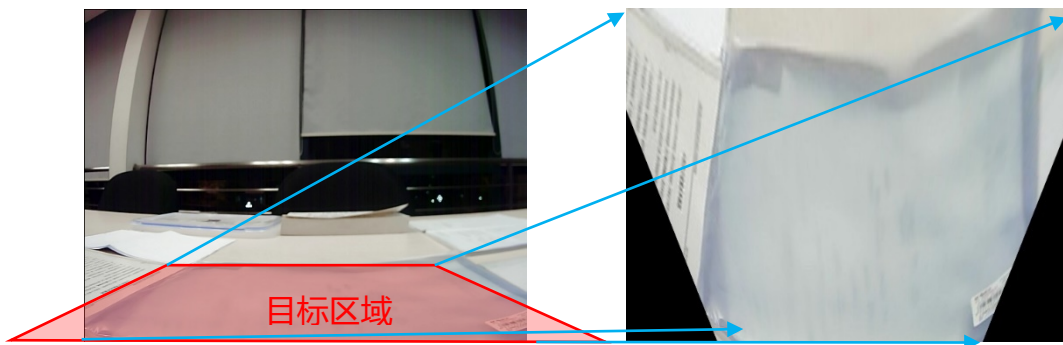


图 2 透视校正示意图

进行透视校正之后，我们在校正图像的基础上再进行灰度化与二值化操作，即可得到一幅只有黑白两色的轨迹线图像，轨迹线图像与实际地图形状大致一致。由于黑色的轨迹线与背景的对比很明显，所以二值化之后的图像能够比较好地再现轨迹线的形状。在图 2 中，由于透视校正左下角和右下角是黑色的，所以我们的二值化进行的是反变换。

2.1.2 控制部分

在得到处理后的图像之后，我们需要在其中提取中有用的控制信息并进行控制。由于小车的前置摄像头位于小车车身的中轴线上，因此当小车中轴线刚好与前方轨迹线重合时，说明小车车身端正，且轨迹线平直，小车直行即可。当前方轨迹线弯曲时，也说明小车需要沿着弯曲的方向调整前轮角度。

为了定量表示前方轨迹线的弯曲程度，我们使用扫描的方法。选取图像中固定的一行，

程序不断扫描图像中该行白色点（代表轨迹线）的像素坐标，并求平均值。将求得的平均值与图像中心线的像素值相减，得到偏移误差。偏移误差的符号表示了车位线行进的方向，而误差的绝对值大小即代表了前方轨迹的弯曲程度。在我们的实际项目中，摄像头返回图像宽 640 像素，高 480 像素。我们不断扫描第 400 行像素的灰度值，找到白色轨迹线的中点像素坐标 x ，由于图像中心线像素值为 320，代表了车身的正前方，则可以求得误差 $e = 320 - x$ ，当 $e > 0$ 时说明车辆应该左转，反之则需要右转。

由于我们控制小车是通过赋值给 `steer` 和 `speed` 两个变量实现的，而 `steer` 与偏差的大小呈正相关，所以我们通过 PID 来控制 `steer` 的值，实际上我们只使用了 P 参数，并且根据弯曲程度的不同使用了两套不同的参数，具体为：当 $|e| > 50$ 时，比例系数为 0.005；当 $|e| < 50$ 时，比例系数为 0.008。在速度上，我们命令当 $|e| > 50$ 时 `speed` 为 0.2，以较慢速度行进，防止小车冲出轨迹线；当 $|e| < 50$ 时 `speed` 为 0.25，以较快的速度前进，尽可能缩短巡线所需时间。

需要注意的是，由于黑色轨迹线会有偶尔的反光现象，发生反光时，我们所扫描的行可能恰好找不到轨迹线。在这种情况下，我们命令程序扫描第 405 行、第 410 行……直到能够找到存在轨迹线的一行为止，这样就能大大提升程序的鲁棒性，降低噪声带来的干扰。另外，由于硬件的原因，我们小车的前置摄像头并不是完全正对中轴线的，当小车实际在轨迹线正中心时，图像上的轨迹线并不在正中心，所以我们设置了 `offset` 抵消硬件的偏移，`offset` 的值大约为 30。

2.2 入库任务的实现

2.2.1 图像处理部分

倒车入库任务中，车辆的后方有四个并列摆放的车库，从左到右分别标号 1~4。我们的小车需要依次倒车进入 1~4 号车库，在倒车过程中不能压线，并且车身要尽可能与车库保持平行。车库的示意图如图 3 所示。

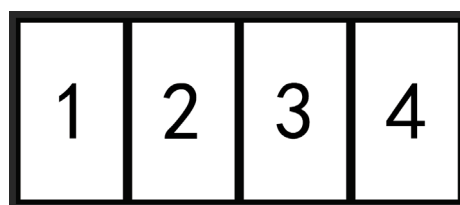


图 3 车位分布示意图

与巡线任务一样，我们首先对图像进行畸变校正，与巡线中不同的是，后置摄像头有非常明显的桶形畸变，因此单纯依靠透视校正是行不通的，得到的图像形变非常明显。需要在透视校正之前对镜头本身的桶形畸变进行校正。没有进行桶形校正之前，后置摄像头读到的原始图像与透视校正之后的图像如图 4 所示，从图中可以发现，提取关键信息是非常困难的，这进一步说明了桶形畸变校正的必要性。

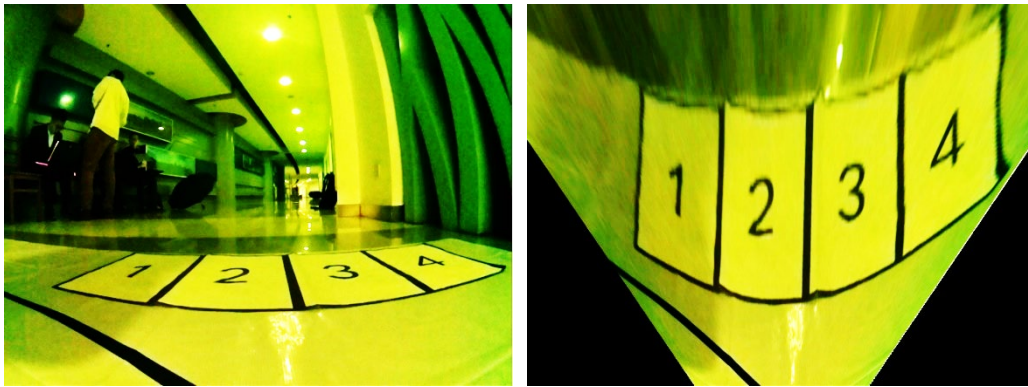


图 4 未进行桶形畸变校正时的原始图像和透视校正图像

经过查阅资料发现，桶形畸变是由摄像头硬件产生的，我们需要用标准标定板进行标定，并使用恰当的工具计算校正矩阵。在 MATLAB 中恰好有镜头校正工具包 cameraCalibrator。具体操作方法介绍如下。首先打开 MATLAB 中内置的标定板显示在电脑屏幕上，开启小车后置摄像头并以不同的角度拍摄标定板，保存这些图片。将图片输入到 cameraCalibrator 中，程序即可自动给出校正矩阵。在 python 代码中调用相关的校正函数并输入校正矩阵，即可输出校正之后的图像。标定照片、工具包界面如图 5 所示。

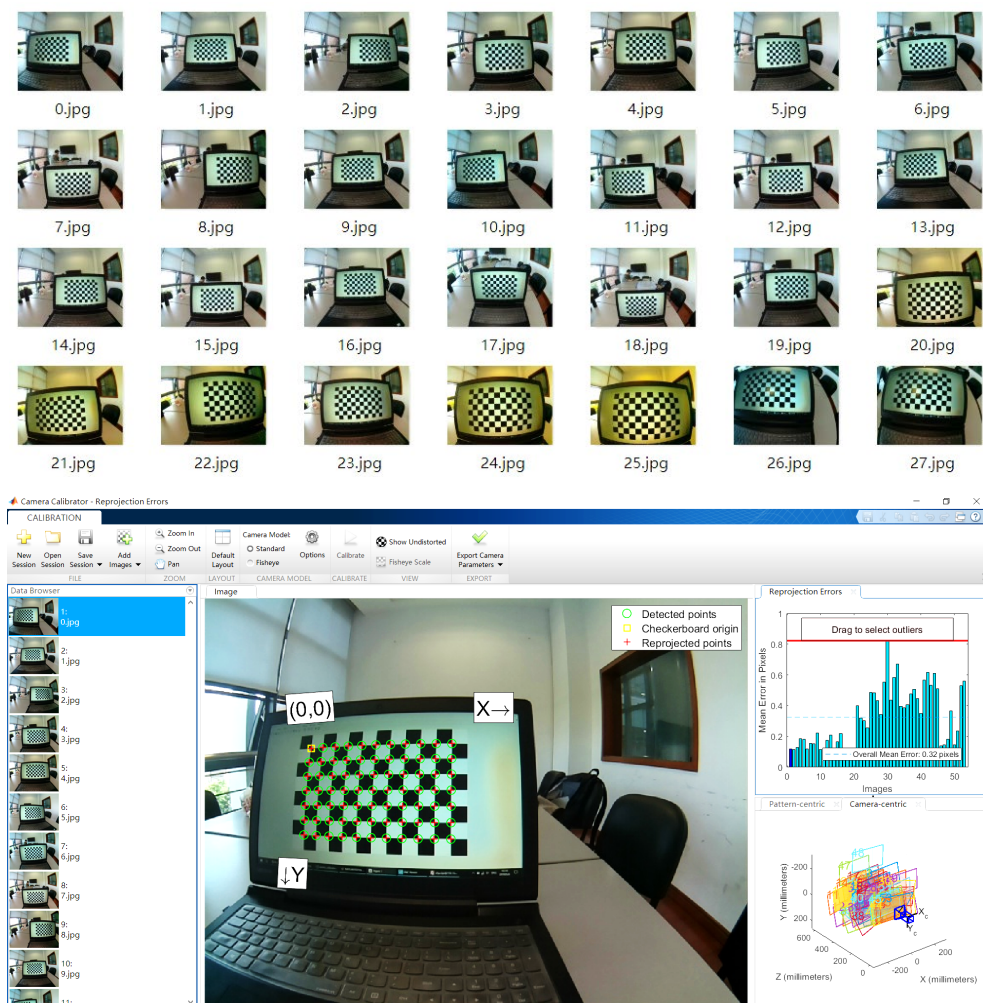


图 5 标定界面示意图

在获得校正矩阵之后，我们即可利用参数得到无桶形畸变的图像，这对我们后期的图像处理非常关键。将桶形畸变校正前和校正后的图像进行对比，如图 6 所示。从图中可以看出，校正过后车位边界线重新恢复平直，我们可以在此基础上再进行进一步处理，从而提取关键信息。

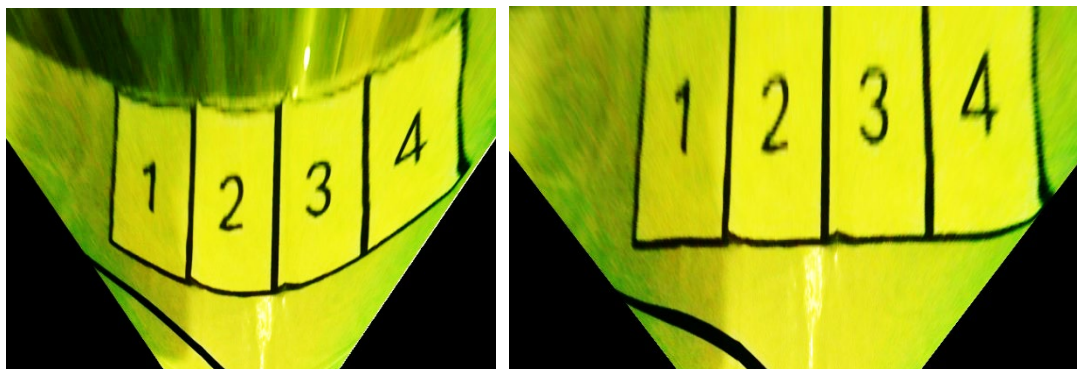


图 6 桶形畸变校正前后的图像，左侧为校正前，右侧为校正后

对图 6 中的右侧图像进行进一步处理。首先将其灰度化、二值化，得到二值图像。我们获得车位位置的思路是找到车位中心的 1、2、3、4 四个数字在图像中的位置。为了准确提取这四个数字，我们采用了轮廓查找的方法，即使用 opencv 内置的轮廓查找函数，通过轮廓的长度准确分离四个数字（四个数字外轮廓的总长度与图像中其余轮廓的长度有明显区别，所以可以通过限定轮廓长度范围找到四个数字的位置），并通过内置函数计算轮廓质心的像素坐标。具体如图 7 所示。在图 7 中，蓝色闭合曲线就是程序找到的轮廓（已经滤除了非数字轮廓），绿色点就是程序计算得到的轮廓质心，它可以表示车库中心点的位置。

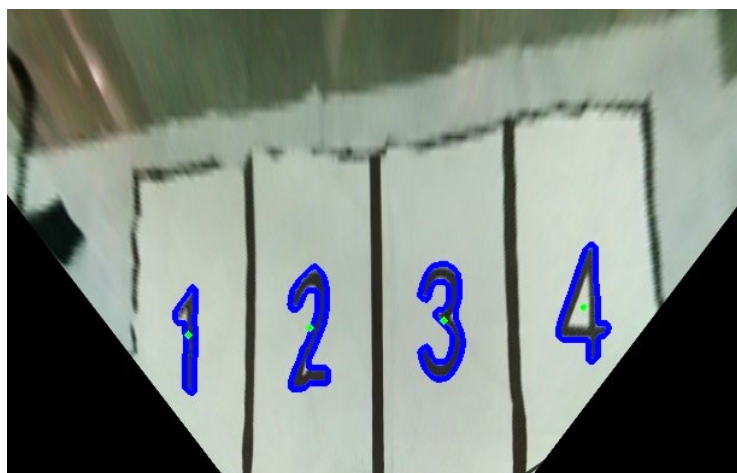


图 7 通过轮廓查找函数确定车库位置

至此，我们就完成了倒车入库图像处理的全部工作，在接下来的一节中，我将详细介绍如何利用查找到的车库中心点坐标对小车进行运动控制。

2.2.2 控制部分

本节中我将详细介绍如何利用图像处理得到的车库中心点坐标确定小车动作，本节中假设小车当前意图进入 2 号车位。

在控制逻辑的设计上，我们采用了“闭环加开环”的方法，基本逻辑如图 8 所示。小车首先从后置摄像头获取一幅图像，将图像中的信息与自己的目标进行比对，从动作函数库中选取相应的动作函数执行。在执行动作函数的过程中，小车不再使用后置摄像头，仅仅根据传入动作函数的参数开环地执行相应动作。在该动作函数执行结束之后，小车再次获取图像并循环执行上面描述的过程。依次倒车进入 1~4 号车库后停车。所谓动作函数，就是小车进行调整的主要方式。比如小车偏离车库中心线时，其首先根据图像计算偏离的横向误差，将这个横向误差传输到一个调整误差的动作函数中进行误差校正。这个函数执行完成之后再次观察图像，如果误差仍然超出允许的范围，则再次进入这个动作函数，否则执行其它的动作函数。接下来我将具体介绍我们设计的各个动作函数及其调用条件。

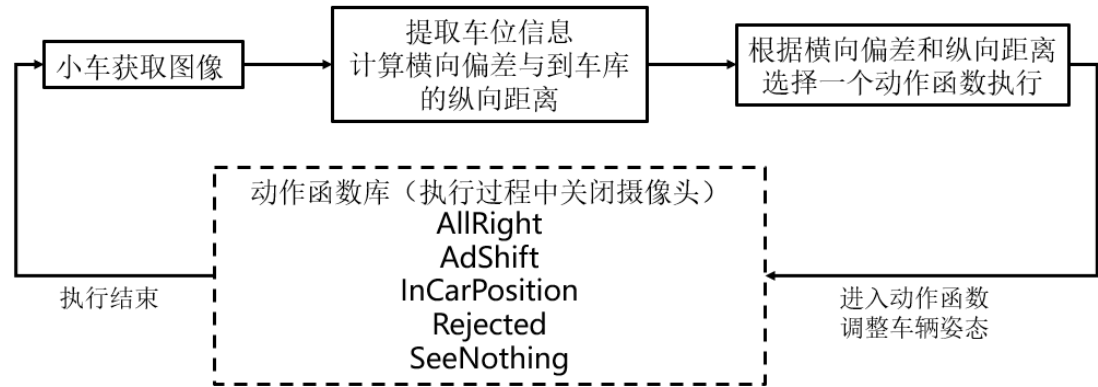


图 8 倒车入库控制基本逻辑

第一种情况。当小车刚好在目标车位的前方且无横向偏差时，小车仅仅需要直线倒车即可。这时候我们在主函数中先计算目标车位与中心的偏差，记为 e （计算方法与巡线中的方法一样），当 $|e| < 20$ 时，判断小车基本对准车位，此时进入 AllRight 函数，表示当前姿态正常，小车可以正常倒车。在 AllRight 函数中仅仅需要命令小车以恒定速度直线后退，并延时 0.5 秒。即函数执行 0.5 秒后退出，进入下一个判断周期。

第二种情况。当小车偏离车位中心线，即 $|e| > 20$ 时，小车需要通过转向校正到车库的正前方，此时我向 AdShift 函数中传入 e ，该函数通过 e 的符号和绝对大小判断需要调整的方向以及调整程度。假设小车在车库的左侧，调整策略是：小车先使前轮向左并倒车，延时一段时间后小车再使前轮向右偏移同样的程度，并再延时同样的时间。如果小车在车库右侧，则前轮先左偏再右偏。初始偏移误差越大，延时时间越长即可。函数执行完成后小车的运动轨迹如图 9 所示。

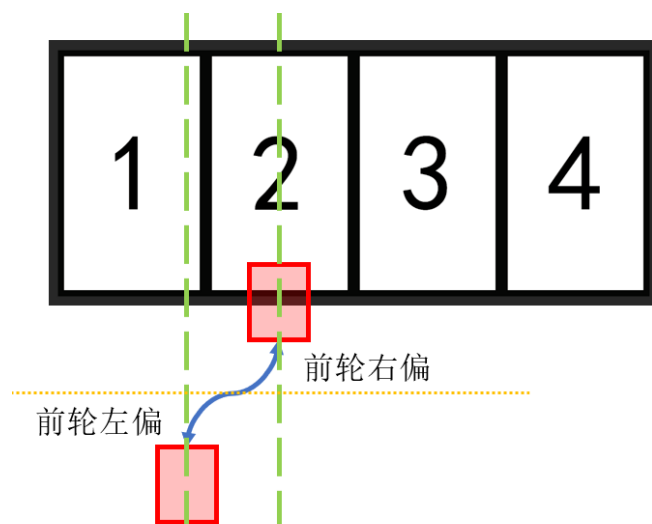


图 9 AdShift 函数的小车进行轨迹

第三种情况，当小车发现数字非常接近图像底部，即小车已经非常接近车库入口时，小车会进行入库前的检查。如果此时小车横向偏差在允许的范围内，那么小车就会执行入库动作函数 InCarPosition，该函数执行了一个一体化动作，它能控制小车按照图 10 所示的轨迹行进到下一个车库的前方。

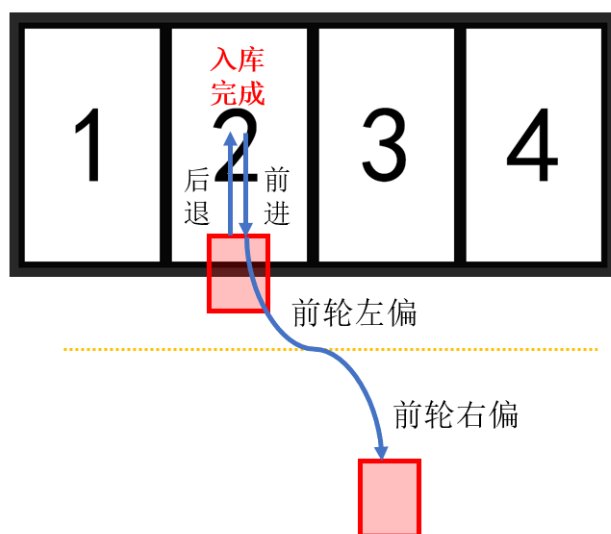


图 10 InCarPosition 函数的小车进行轨迹

第四种情况承接第三种情况而来。当小车在靠近车库时发现车库中心与车体中心偏差过大，有压线风险，那么小车会自动放弃倒车入库，执行 Rejected 函数，转而向前直行，留给小车充足的空间再次调整横向偏差。该方法极大提升了我们系统的稳定性，避免了小车压线的风险，在我们提交的视频中，即可显示出这种控制方式的优良特性。

第五种情况是为了提升鲁棒性设置的。小车有时会受到环境影响而对某些数字漏识别，有时因为函数延时的原因而超出图像识别的界限，这些都会导致小车看不到数字，所以我们必须让小车知道，在未能成功识别到数字的情况下如何执行动作。SeeNothing 函数就是为

此情况设计的。执行该函数时，小车首先会读取前一次执行的动作函数种类，以此判断小车位于车位的前方还是后方。如果位于车位前方，则直线倒车，否则直线前进，知道再次看到数字为止。

以上介绍了我们控制小车的全部逻辑，经过实际测试发现效果良好，特别是小车在极端情况下能够自行判断调整，鲁棒性让我们满意。需要说明的是，我们没有考虑车身倾斜的情况，原因是在每个动作函数执行完毕后，我们都能保证车身与车库中心线平行，没有再实现倾斜调整的必要。事实上，我们也实现过倾斜调整函数 `AdTilt`，但是此函数从未被小车调用，除非初始放置时车身倾斜就很严重，故在本文中不再介绍。

3 研究成果

在本节中，我将结合开题报告中的预期计划和任务目标，详细介绍我们小组的项目成果。本节主要分为时间进度和任务完成情况两个部分进行说明。

在时间进度方面，我们预期在第 3~8 周完成巡线任务及其附加要求，在第 9~13 周完成倒车入库任务及其附加要求，在第 14~16 周进行算法调整和鲁棒性提升的工作。实际进度中，我们与预期相差不大，在第 9 周时，由于倒车入库任务进入了瓶颈期，多次尝试都没能实现任务要求，所以进度停滞了两个多星期。第 14 周左右我们基本完成了所有任务要求，进度渐慢，主要进行一些微调的工作。其余时间我们每周都会对小车进行大量调试，渐进式地完成任务要求。

本课程以树莓派和后驱四轮小车为平台，基于 RGB 摄像头取得图像，对图像进行处理和信息提取，最后基于任务要求对小车进行相应控制。接下来我将分别介绍巡线任务和入库任务的研究成果。

在巡线任务中，我们能够控制小车在 20 秒左右走完测试轨迹（一圈），小车在巡线过程中转向稳定，车辆始终位于黑色轨迹线之上，中轴线与轨迹线偏差较小且能够自动调节。我们不断读取摄像头取得的图像，对取得的图像进行透视校正、二值化与边界提取、降噪等处理，将一个原始的 RGB 图像转化为一张轨迹趋势的俯视二值图。通过读取特定位置轨迹线与中轴线的偏差，我们就能得到小车前方一定位置上轨迹线的走向，并以此控制小车的转向和速度。为了尽可能减少小车巡线的时间，我们命令小车在轨迹弯曲程度大时放慢速度，而在平直的路段以较快的速度前进。依据以上逻辑，我们最终完成了整个巡线任务，在我们花费的巡线时间比要求短得多，我们巡线的精度也比要求高很多，总得说来我对巡线任务的完成情况比较满意。

在倒车入库的任务中，我们能够控制小车依次倒车进入 1 号~4 号车位，并且在倒车过程中能够根据后置摄像头返回的图像实时调整偏移程度。当小车靠近车位即将入库时，小车能够进行入库前的检查，防止车体以横向偏移和倾斜的姿态进入车库。当车体姿态不满足入库要求时，小车放弃入库，前进一段距离重新调整，直到车身与车位线近似平行且不压线。

为了实现上述功能，我们首先利用 MATLAB 软件和标准标定板，对后置摄像头的桶形畸变进行了校正，随后我们又在校正后图像的基础上进行了透视矫正、二值化、轮廓查找与筛选等工作，提取出了车位中心数字的坐标，利用这些坐标判断车位与小车的相对位置。依据以上逻辑，我们最终实现了小车依次倒车进入 1 号~4 号车位的功能，倒车过程平稳，前轮无抖动。车辆全程不会压线，且倒车进入车库后车身基本与两侧边线平行。我们对倒车任务的完成情况基本满意，美中不足的是小车在极少数情况下会产生数字识别遗漏的问题，这会对小车的动作执行造成影响，经过调整优化，这种情况出现的概率已经非常低了。

总的来看，我们小组完成了所有的任务要求，并在原有要求的基础上增添了一些鲁棒控制的功能，使得运行更加稳定、某些性能指标远远超出预期值。除了每个任务的基本要求之外，我们还完成了全部的附加要求。

4 研究总结

4.1 研究中的问题与讨论

课程项目进行的过程中，一些问题常常会给我们的任务实现带来困难，在尝试解决这些问题的过程中，我们曾经使用了多种方法，并根据实际效果进行了取舍。在第二部分介绍的方法仅仅是我们最后使用的、效果较好的方法，在本节中，我将较为简略地介绍我们曾经尝试过的技术途径，并说明舍弃这些方法的具体考虑。同时，我还将总结一些关键技术的使用考虑。

第一。在整个任务过程中，摄像头视角会给信息提取带来诸多困难。从小车上前后摄像头放置的位置可以看出，小车是平视观察前方的，但是我们需要关注的仅仅是地面上的信息，但地面仅仅占据图像的一小部分。如果对图像的视角不做任何处理，那么地面以外的信息（如行走的人、窗外的复杂环境光等）就会对关键信息提取产生巨大的干扰；地面上较远位置的物体也会由于透视原理而显得非常小，不利于我们的图像处理。我们想到，如果能把图像校正成为看向地面的俯视图，那么既可以去掉大量干扰，又可以将被观察物体的透视形变校正过来（去掉“近大远小”现象），方便调试和信息提取。所以我们在网络上查找到了透视校正的方法，并根据我们摄像头的实际情况进行了调整，最后得到了非常漂亮的俯视图，具体方法在第二部分已经介绍。事实证明，透视校正的应用为我们之后的图像处理工作打下了良好的基础。

第二。在巡线和倒车入库的过程中，我们发现两个摄像头都有明显的桶形畸变，其中以后置摄像头最为明显。桶形畸变的存在非常影响我们倒车入库的实现，因为我们曾尝试通过直线判断车位的位置，如果畸变存在，那么直线就会变得弯曲，程序无法识别。查阅资料发现桶形畸变是由于摄像头硬件导致的，可以通过标定的方法加以校正。我惊喜地发现 MATLAB 软件有内置的镜头畸变校正工具包，我们仅需要让被校正摄像头对着标定板拍照，软件就能自动通过拍得的照片计算得到校正参数矩阵。具体实现方法已在第二部分介绍。通过实际应

用，我们发现畸变校正的效果非常好，在喜悦之余，我也对 MATLAB 的强大功能表示惊叹。这次的经历也让我对摄像头桶形畸变和枕形畸变有了直观的认识，也让我知道了一个校正畸变的快捷途径。

第三。在巡线过程中，我们一开始尝试了非常复杂的控制方法，效果并不理想，最后我们对控制方法进行简化，控制效果反而变得不错。我们最初的想法是，在最后的二值图像上选取几行，分别代表距离车体不同距离的轨迹线横向位置，给靠近车体的偏差较大的权重，给远离车体的偏差较大的权重。这种方法虽然给与小车预测轨迹走向的能力，但是在实际测试中效果并不好，小车经常冲出轨迹线。经过思考我们发现，“预测”对小车来说是不必要的。比如在走 S 形连续弯道时，小车远离车体的探测点可能已经到了下一个弯道的位置，这会给小车一个错误的转向命令。我们最后使用了一种非常简单的方式，直接观察前方一定距离的一个观测点，只通过这一个观测点与中心的偏差来决定小车的走向，这种方法非常直接，但是效果优异。由于采用了透视校正，环境噪声对图像的影响也不大，最终达成了较好的控制效果。

第四。在倒车入库的过程中，我们尝试了多种方法，试图提取车库与小车的相对位置，但都无果而终，这也是我们项目过程中遇到的最大的困难。我们一开始尝试通过识别直线获取车库位置，但是霍夫变换方法无法去除噪声，直线识别结果非常不稳定，一些边界线时有时无，我们无法根据直线信息进行控制。随后我们改用模板匹配的方法，尝试识别出边框交界处的位置，以此来推断各边框中心的相对位置。但是模板匹配过于“死板”，当环境光线变化、小车远离和倾斜时，关键点会出现形变，模板匹配因此失效。我们最后尝试使用轮廓线识别数字的方法，因为车位中心的数字轮廓线明显长于噪声的轮廓线，我们可以根据轮廓线长度找到数字的位置，以此来判断车库位置。实际测试中，这种方法效果最好，所以我们最后得以保留。

第五。在倒车入库的过程中，我们需要使用延时函数。具体实现方法是直接调用系统的 sleep 函数。但是延时函数的使用不仅会造成摄像头返回图像的延时，也会让电机的时间控制不精确，小车不受控。查阅资料和咨询助教后，我们最后使用了一种非常巧妙地办法。我们自己实现了一个延时函数，进入延时函数时取系统的当前时间并进入 while 循环，每次循环都获取摄像头图片并获取系统当前时间，当时间差满足要求时退出循环。这种方法的好处在于，我们延时的过程中摄像头是一直工作的，图像获取是不会中断的，所以每次小车进行动作判断时都会拿到当前的即时图像，而不是延时之前的图像。这种方法保证了小车计算的实时性，使得我们的控制更加高效。

第六。在倒车入库的实现中，我们采用了“闭环加开环”的方法，具体实现已经在第二部分中详细介绍。之所以采用这种方法，是我们考虑到如果小车是完全基于图像实时判断的，那么当小车进入调整状态时，摄像头传来的信息事实上成为了一种“干扰”，因为调整中的小车看到的车位是不完整而且倾斜的，如果我们也处理这种情况，那么整个系统将会变得极

其复杂，容易出错。因此我们将小车的动作分解成为一个个小节（即动作函数），每一小节内部是开环的，小节与小节之间是闭环连接的，这样既获得了开环控制简洁的优良特性，又能够使小车有充分的调整能力，一举两得。另外，由于采用了部分开环控制，我们能够保证小车每次执行完动作函数之后保持与车位平行，因此小车全程不会倾斜，我们也免去了校正倾斜误差的麻烦。

4.2 感想与体会

在一个学期的课程实践中，我们小组成员齐心协力、各有分工。在巡线部分，我主要负责图像处理和信息提取的工作；在倒车入库部分，我则主要负责小车控制逻辑部分的代码。小组成员的工作组合起来，共同完成了各项任务要求。在整个过程中我们对课程学到的知识充分加以利用，并结合工程要求，广泛查阅资料，在解决实际问题的过程中锻炼了各项能力。虽然最后结果符合预期，但是过程不总是一帆风顺的，我在这个过程中收获了很多知识层面和经验层面的东西，在此略作总结。

在这门课中，通过摄像头获取的数据控制小车，我们达到了巡线和倒车入库的目的。联系到当今火热的自动驾驶领域，我突然发现本课程的工作其实就是实际自动驾驶的简化，并且还涉及到自动驾驶的热门研究方向和行业应用（如倒车入库、辅助驾驶等）。这让我从内心对课程项目有了浓厚的兴趣，也让我体会到了本门课程的必要性。机器视觉领域是当今的热门研究领域，本门课程对视觉处理的大量应用，为我打下了良好的应用基础。

实践过程中，我还体会到了理论知识与实践的差别。以图像处理为例，在课程中我们虽然学习了很多经典的图像处理方法，但是在我们实际调试小车时才发现环境中的噪声很难处理，比如逆光直射、地板反光、遮挡产生的阴影等，一些形态学的处理也很难进行，特别是在无法利用神经网络等人工智能算法的情况下。我们在实际情况中，往往是在理论的基础上对各种方法进行加工改进，使得整个算法适用于我们的小车系统。在其它工程问题中，这种情况也是普遍存在的，工科创 3F 课程让我对工程实践有了更加深入的认识。

在项目进行的过程中，有进展快速的时期，也有瓶颈期。探索的过程中存在着诸多未知的因素，对待工程任务的心态非常重要。我们小组在倒车入库的图像处理过程中，多次尝试提取车库未知信息，但都以失败告终。我们曾尝试通过霍夫变换提取直线，但是最终有很多误识别的直线，信息无法有效利用；我们也曾尝试过模板匹配的方法，但是识别精度不高，会有漏识别现象。那几个星期我们常常冥思苦想但不得其解，心态很崩溃。但是当我们冷静下来，跳出之前的死胡同，与其它同学沟通交流之后，问题得到了很快地解决。这说明工程难题是家常便饭，我们不能一味消极应付，用合适的方法、独特的眼光和积极的态度，才是解决问题的正道。这门课程为我之后学习和工作提供了一些启示。

最后，我还想感谢我的两名队友，在与他们合作的过程中，我不仅完成了复杂庞大的工程任务，还锻炼了团队协作能力和沟通能力，知识和经验的相互交流更是让我学到了课本上没有的东西。

4.3 建议与致谢

课程运行中，助教及时有效地发布各项通知，及时解答同学们的疑惑，因此我认为在课程秩序方面已经非常完善了。在任务进行方面，我建议老师和助教更早地完成赛道布置和指定的工作（可以在开题时就明确指定巡线赛道形状），这样我们就能更早在实际赛道上调试和测试。在任务验收方面，我十分建议同一小组共同完成一份结题报告，而不是一人提交一份。因为一个小组完成的工作是一样的，当小组成员共同完成一项任务时，报告的内容都是大致相同的，大家对技术问题交流很多，重复撰写多次没有意义；当小组成员分工非常明确时，大家又对其他成员的内容不太了解，写报告也会产生不完整的情况。如果老师本意是想通过写多份报告的方式测试成员的参与度，我觉得可以采用组员匿名互评的方法，参与度不高的成员自然会在分数上有所体现。除此之外，我认为本课程在其余所有方面都非常优秀，在保证趣味性的同时，也让我们学到了不少知识。

课程进行的过程中我们遇到了很多困难，咨询助教时总能得到助教的及时和耐心的帮助，助教也为我们的一些技术细节进行了提示，实实在在地帮助我们解决了一些棘手的难题。在此我想特别向课程的老师和助教表达感谢！另外，我们还与其它小组的同学进行了深入的交流，在此过程中也得到了一些难得的启发，在此也向本课程的其他同学表达感谢！作为小组的成员之一，我们小组在整个学期中互帮互助，高效地完成了所有任务，三位成员都很努力，在此我也向我的队友赵寅杰和严威豪表达诚挚谢意！