

学校代号 10532

学 号 S12021112

分 类 号 TP391

密 级 公开



湖南大学
HUNAN UNIVERSITY

硕士学位论文

基于图像识别的智能小车路径跟随 系统设计

学位申请人姓名 周南

培 养 单 位 机械与运载工程学院

导师姓名及职称 宋晓琳 教授

学 科 专 业 机械工程

研 究 方 向 车辆与交通安全

论文提交日期 2016 年 5 月 10 日

学校代号:10532

学 号: S12021112

密 级:公开

湖南大学硕士学位论文

基于图像识别的智能小车路径跟随 系统设计

学位申请人姓名：周南

导师姓名及职称：宋晓琳 教授

培 养 单 位：机械与运载工程学院

专 业 名 称：机械工程

论文提交日期：2016 年 5 月 10 日

论文答辩日期：2016 年 5 月 26 日

答辩委员会主席：程军圣 教授

Design of intelligent vehicle path following system based on
image recognition

by

ZHOU Nan

B.E.(Hunan University)2012

A thesis submitted in partial satisfaction of the

Requirements for the degree of

Master of Science

in

Mechanical engineering

in the

Graduate School

of

Hunan University

Supervisor

Professor SONG Xiaolin

May, 2016

湖南大学

学位论文原创性声明

本人郑重声明:所呈交的论文是本人在导师的指导下独立进行研究所取得的研究成果。除了文中特别加以标注引用的内容外,本论文不包含任何其他个人或集体已经发表或撰写的成果作品。对本文的研究做出重要贡献的个人和集体,均已在文中以明确方式标明。本人完全意识到本声明的法律后果由本人承担。

作者签名: 周南

日期: 2016年 6月 6日

学位论文授权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定,同意学校保留并向国家有关部门或机构送交论文的复印件和电子版,允许论文被查阅和借阅。本人授权湖南大学可以将本学位论文的全部或部分内容编入有关数据库进行检索,可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

本学位论文属于

1、保密□,在_____年解密后适用本授权书。

2、不保密□。

(请在以上相应方框内打“√”)

作者签名: 周南

日期: 2016年 6月 6日

导师签名:

日期: 2016年 6月 6日

摘 要

汽车智能驾驶系统要实现车道线检测、障碍物检测、路径规划及路径跟随等几项基本功能。路径跟随控制作为智能车辆研究的关键问题，近年来受国内外学者的广泛关注。本文研究基于图像识别的路径跟随控制算法，并以 STM32F103 为核心控制模块，搭建 20:1 的嵌入式智能小车模型，通过实验对路径跟随及其控制效果进行验证。

论文主要内容包括以下几个方面：

(1) 根据智能小车路径识别及路径跟随的需求，设计智能小车系统。以 STM32F103 开发板为开发平台构建嵌入式硬件控制系统，包括局域网摄像头模块、蓝牙串口通信模块、电源模块、电机驱动模块、测速模块及各个模块间的接口电路设计。

(2) 研究智能小车实现路径跟随的算法。为了高效稳定地实现智能小车路径跟随，避免复杂的运算，借鉴模糊控制算法的定性输入思路，根据路径走向偏离中线角度的大小划分集合，根据所在集合输出小车双轮转速以实现转向，采用快速调整行车方向和行车方向微调相结合的方法，实现智能小车路径跟随。

(3) 研究了智能小车电机的 PID 调节方法，以及小车车速的测量方法。智能小车为三轮模型，包括两个驱动轮和一个万向轮，转向由双驱动轮差速旋转实现。电机调节采用闭环 PID 调节，测速采用适用与低速测量场合的测周期法。

(4) 通过仿真实验将本文算法与传统的通过最小二乘法拟合路径中心线获取路径信息的预瞄控制算法作对比，比较两种算法的计算量和准确度。使用搭建的智能小车进行实验，结果表明，小车可以按给定路径稳定行驶。这证明基于 STM32F103 开发板所建立的嵌入式平台，及基于图像识别的路径跟随算法是正确可行的。

关键字：智能小车；路径跟踪；图像识别；模糊控制

Abstract

Intelligent vehicle driving system includes several sub system such as lane detection, obstacle detection, path planning, path following and so on. Path following is a key part in the research of intelligent vehicle, which is widely concerned by domestic and foreign scholars in recent years. This paper proposes a path following control algorithm based on image recognition. And a 20:1 intelligent vehicle model with STM32F103 chip as the core of control module, was designed to accomplish the path following research and its control method research. Based on experiment, the path following algorithm and its control effect were validated.

The main contents of this paper are as follows:

(1) Designing the intelligent car system and building the vehicle model, according to the requirement of intelligent vehicle path identification and path following. The vehicle model controlled by a STM32F103 chip board is the hardware part of the intelligent vehicle system. It is formed by camera module, Bluetooth serial communication module, power supply module, motor drive module, velocity measure modules and circuit connection between the modules.

(2) The path following algorithm for intelligent vehicle system. In order to make the vehicle follow the path efficiently and stably, this paper divide the angle of path deviating from right ahead into several sets. Each set corresponds to a running mode. And this is so-called qualitative input, learning from fuzzy control algorithm. The path following control is realized by the combination of coarse and fine adjustment of the direction of the running vehicle.

(3) The velocity measure method and the PID control method of the intelligent vehicle motors. The vehicle model is a three-wheel model, comprising two driving wheels and an universal wheel. Two driving wheels rotate at different speed to make the vehicle turn direction. The PID closed-loop control method is used to control the motors, and the measuring period method, which is widely used for the measurement of low speed, is applied to the measurement of the wheels.

(4) Through the simulation experiment, this paper makes a comparison between the algorithm and the traditional preview control algorithm which can get the path information by the least square method, and compares the computation and the

accuracy of the two algorithms. The result of the experiment doing by the model of the vehicle shows that the vehicle can be driven by the planning path. This proves that the embedded platform based on STM32F103 board, and the path following algorithm based on image recognition is correct and feasible.

Keywords: Intelligent vehicle; path following; image recognition; fuzzy control;

目录

学位论文原创性声明和学位论文版权使用授权书	I
摘 要	II
Abstract	III
第 1 章 绪论	1
1.1 研究背景及意义	1
1.2 路径图像识别研究现状	4
1.3 本文研究目标	5
1.4 本文组织结构	6
第 2 章 智能小车体系结构与建模	7
2.1 智能小车系统结构	7
2.2 智能小车运动学建模	9
2.2.1 智能小车的坐标系	9
2.2.2 智能小车位姿描述	10
2.3 本章小结	11
第 3 章 图像处理与摄像机标定	12
3.1 图像处理基本原理	12
3.1.1 图像灰度化	12
3.1.2 最优阈值分割	12
3.1.3 数学形态学操作	14
3.2 二维平面的投影变换及标定方法	15
第 4 章 路径特征提取与路径跟随控制	18
4.1 方向码盘	18
4.1.1 方向码盘概念的提出	18
4.1.2 方向码盘透视变化	19
4.2 模糊控制器的设计	20
4.2.1 输入量与输出量的确定	21
4.2.2 模糊规则	22
4.3 小车运动控制	23
4.3.1 智能小车指令执行时间	23
4.3.2 小车左右轮速度	25

4.3.3 小车偏航复位	25
4.4 模糊控制规则的补充	26
4.5 上位机 (PC) 控制软件功能设计	27
4.5.1 上位机 (PC) 软件开发环境	28
4.5.2 图像接收处理的实现	29
4.5.3 串口数据收发	31
4.6 实验仿真	31
4.7 本章小结	33
第 5 章 智能小车路径跟随试验	34
5.1 硬件系统布置	34
5.2 智能小车平台搭建	35
5.2.1 上位机配置	35
5.2.2 移动端设计与搭建	35
5.3 小车轮速调节	44
5.3.1 驱动电机调试	44
5.3.2 轮速 PID 调节	46
5.4 指令延时及方向修正	47
5.5 实验	49
总结与展望	51
1 主要的研究工作和创新点	51
2 工作展望	51
参考文献	53
致 谢	57
附录 A 作者在攻读学位期间发表的论文目录	58
附录 B 实验数据	59

第 1 章 绪论

1.1 研究背景及意义

自汽车问世以来，历经百余年的发展，汽车早已成为人们日常生活中不可或缺的一部分，它的发明已成为人类现代文明进步的标志。随着汽车保有量的日渐增加，我们在享受汽车为彼此生活工作带来便利的同时，也为此付出了沉重的代价。据相关数据统计，全球平均每分钟至少有一人因交通事故而丧命^[1]。

世界卫生组织的研究结果表明，因道路交通事故死亡的人数远多于战争，与此同时还伴随着巨大的经济损失。据相关数据统计，全球每年有近 130 万人死于道路交通事故，经济损失更是高达 5180 亿美元^[2]。在我国，1951 年至 2015 年，60 年期间全国机动车的总保有量已达到 2.79 亿辆，累计已有 2231148 人死于交通事故。其中，仅 2015 年一年时间，全国共发生了 21017 起交通事故，死亡人数达到 72387 人^[3]。我国俨然已步入交通事故发生频率最高的国家之列。

面对日益严峻的道路交通安全问题，各国政府与广大研究机构都对此高度重视，制定相关的研究计划的同时还加大了对研究解决方法的投入。减少道路交通事故、提高汽车安全性能等问题早已成为人们日常关注的社会热点，更是科学技术进步所面临的重要课题^[4]。国内外研究机构通过研发被动安全技术，如安全带、安全气囊以及吸能式车身结构等，来减少因交通事故而造成人员伤亡和财产损失，但由于没有有效解决引起交通事故的根源性问题，大量交通事故仍然无法避免。因此，在汽车发生危险前，提醒驾驶员或在紧急状况下，辅助驾驶员采取相应的安全措施的各种主动安全技术备受关注并得到迅速发展。

上世纪 80 年代以来，随着计算机技术、电子传感器技术、自动控制技术、信息融合技术以及图像处理技术等飞速发展，将这些技术如何运用于汽车领域，从而降低交通事故的发生频率，减轻驾驶人员的驾驶负担，也已成为全球备受关注的研究课题。例如：兴起于美国的智能交通系统（Intelligent Transport Systems）^[6]，不仅是这些技术运用于汽车领域的典型体现，同时也是解决各类交通安全问题的重要途径。ITS 主要由智能车辆系统（Intelligent Vehicle Systems）和智能道路及交通控制系统组成^[8]。目前，智能道路系统的构筑仍然处于起步阶段，而相应的投资十分巨大，且基础设施建设周期长。因此，减少交通事故发生率的优先选择无疑是发展智能车辆系统。美国早已于 1998 年便开始研究以主动防撞系统为

核心的智能车辆系统，并取得了阶段性成果。

智能车辆是一个集中运用了信息融合、现代传感、计算机、通讯、自动控制技术、以及人工智能于一体的综合系统。作为智能车辆的微缩模型的智能小车，是一个集行为控制与执行、动态路径规划与决策以及环境感知的智能系统。它能验证智能车辆的控制算法，实现小车自主导航、避障、路径跟随控制，模拟真车行驶状态，对于智能车辆的研究有一定的辅助作用。

现阶段，智能车辆系统研究的主要辅助方法为对地面移动机器人的研究。智能车的跟踪问题可以分为三类：点镇定、轨迹跟踪和路径跟随。其中，点镇定即从一个点到达另一个点，只需到达目的地即可，而对过程中走的路径并没有任何要求，这个过程不限定具体的轨迹和时间，研究者可以根据自己的需求定义点镇定是路径最优、时间最优还是路径和时间两者都最优。轨迹跟踪即指定一个与时间有关的曲线，要求载体在一定时间达到指定曲线上的期望点，这要求在空间与时间上保持一致，约束条件比较强。如果中途出现了一些突发状况，那么，移动机器人的轨迹跟踪就可能会出现中断，这时就需要借助手动干预来完成，因此，轨迹跟踪相对而言是一个前向的开环系统，并且鲁棒性不强。路径跟随则是移动机器人跟踪一个与时间毫无关系的几何曲线，要求移动机器人能到达指定曲线上的最近点就行，操作起来没有时间上的约束，它首要强调的任务是到达位置，而考虑机器人本身的机动性则是其次。综上，路径跟随更有实际应用价值。

纵观智能汽车的研究可以概括为三个阶段。

第一阶段（1950-1970），智能车辆研究初始阶段。通过在道路上铺设制导电缆等方式实现横向控制^[10]，其中德国 Siemens 公司、美国的 RCA 项目和 GM、Ohio 州立大学、英国的道路交通研究所和日本的机械技术研究所都开展过相关的研究^[11]。

第二阶段(1970-1980)，智能车辆研究起步阶段。研究重点为基于计算机视觉的智能车自动驾驶系统，该阶段的研究更侧重于对恶劣环境的适应、车辆的速度控制和一些基本控制原理。日本机械技术研究所于 1978 年进行了世界上首次基于计算机视觉的无人驾驶汽车道路试验^[12]，经试验，汽车能以 30km/h 的速度完成简单的循线、导航和避障任务；在 80 年代后期，美国 Maryland 大学开发了用于越野环境下的无人侦察 AL(Autonomous Land Vehicle)^[13]。CMU 开发了 Nav Lab-1 自动驾驶车，它完成传感器信息融合、图像处理、路径规划、图像理解、车体控制，但车辆行驶速度较低；与此同时由富士通与日产联合研制的 PVS 自动驾驶汽车，具有雨天、夜间等恶劣天气下自动驾驶以及自动回避静止障碍物的功能^[14]，德国联邦国防大学（Universität der Bundeswehr München）也开发了 Va Mo Rs 自动驾驶汽车^[15]，并成功将车速提高到 90km/h。

第三阶段(1990 至今)，智能车进入快速发展阶段。这一阶段世界主要发达国

家都针对智能交通系统和军事系统进行了全面而深入的研究工作，以满足交通安全和军事安全的实际需求。在欧洲，以各大汽车厂家和大学为中心的智能车项目均着手研究智能交通系统和驾驶员辅助系统^[16-18]。日本在此期间设立了 ASV 项目、SSVS 项目分别研究了智能车和周边基础设施相协调的自动驾驶系统与电动汽车车间通信自动驾驶系统。美国开展了一系列的汽车自动驾驶系统研究，加州的 ITS 项目 PATH 中，研究保持车距的车队的纵向控制和横向控制^[19]；1997 年，美国交通部在加州组织进行了针对汽车自动驾驶的 AHS（自动化高速公路系统）项目的示范性试验^[20]，之后又展开以车辆安全辅助驾驶为中心的 IVI 项目，已经有一部分研究成果先后应用于货车、巴士、轿车及特种车辆等领域^[21]；CMU 研制的 Nav Lab 自主车系列，全面深入研究了视觉导航技术与基本原理^[22]；90 年代提出了由美国国防部高级研究局（DARPA）负责实施的 DEMO 计划，先后研制了 10 辆自主的实验样车，并在复杂的环境中依照军事需求进行了自主导航技术的研究^[23]。

2000 年后，全球汽车电子展现出巨大市场潜力，并以美国军方无人战争需求为发展的导向和驱动力，智能车在民用和军用领域都处于高速发展阶段。在 DARPA 组织几次越野和城市道路智能车大赛中不断涌现出优秀的智能车设计。CMU 的 Sandstorm 智能车凭借自主行驶了 7.4 英里的战绩取得 2004 年比赛的第一名^[26]，来自 Stanford 大学的 Stanley 智能车获得 2005 年度沙漠越野赛的第一名，CMU 大学的比赛车辆 Boss 以 4 小时 10 分获得 2007 年度城市道路赛的冠军。斯坦福大学于 2009 年 9 月研制出最新的智能车 Shelley，利用 GPS 进行路径跟踪可以实现稳定、高速行驶。

20 世纪 80 年代，我国也开始了智能车的研究。各大高校和研究所纷纷开始研究智能车平台系统，国家启动“863”计划，全面部署智能车技术研究战略^[27]。与此同时，我国各研究机构与大学开始对智能车的相关领域进行研究。西安交通大学研制的 Spring Robot 用 CMU 提供的视频数据和自主搭载的车载摄像头抓取的图像进行道路线识别的研究^[28]；清华大学研制的 THMR-V，对车道线进行识别和跟踪，平均时速可达到 100km/h^[29]；吉林大学研制的 JLUIV 可以在多种环境下进行交通标志符与自身路径的识别以及自动导向和自动避障等的研究，并在一汽大众的捷达轿车变速箱装配线上进行实际生产测试^[30]；2000 年 6 月，国防科技大学研制的 CITAVT-IV 作为第 4 代无人驾驶汽车以 76km/h 的车速创下最新记录。

2009 年 6 月，首届中国“智能车未来挑战”比赛在西安举行。代表国内智能车研制最高水平的清华大学、湖南大学、北京理工大学、上海交通大学、国防科技大学和西安交通大学等 7 所大学的智能车参加比赛。比赛的内容包括特色表演、指定路线的挑战性测试和规定动作测试等，最终，湖南大学与上海交通大学的两支参赛车队脱颖而出，湖南大学获得冠军。

2013 年，“百度无人驾驶汽车”研发计划正式启动。百度公司于 2015 年 12 月宣布，百度无人驾驶车国内首次实现环路、城市及高速道路混合路况下的全自动驾驶。根据百度公布的路测路线显示，百度无人驾驶车从位于北京中关村软件园的百度大厦附近出发，驶入 G7 京新高速公路，经五环路，抵达奥林匹克森林公园，并最终按原路线返回。百度无人驾驶车不仅实现了往返全程自动驾驶，而且还实现了多次超车、调头、变道、跟车减速、上下匝道等复杂驾驶动作，完成了从进入高速（汇入车流）到驶出高速（离开车流）的不同道路场景的切换。测试过程中，百度无人驾驶汽车最高速度达到 100 公里/小时。

1.2 路径图像识别研究现状

智能车辆技术如今已成为各大高校、企业研究机构的热门研究领域，也成为全社会关注的热点。我国智能车技术的发展相对国外而言起步较晚，科研投入有限，因此与国外的技术还存在着一定的差距，还有许多地方有待深入研究。智能车辆是一个交叉学科，其智能化主要体现在车辆的自主性和与外界环境的交互性。这就需要智能车辆具有较好的环境感知能力、路径规划技术和良好的运动可控性。

在环境感知方面，与其他传感器相比，计算机视觉具有其独特的优势，如采集信息量大、能遥测等，缺点是复杂环境下要将探测目标与背景分离开来，需要的图形计算量很大，容易导致系统的实时性较差。计算机视觉从 20 世纪 50 年代的统计模式识别开始，国内外对于智能车辆计算机视觉的研究从未间断，目前，计算机视觉仍然是一个非常活跃的研究领域。随着神经网络、人工智能、信号处理以及其他相关领域学科的发展，计算机视觉的新方法、新理论、新概念不断涌现，计算机视觉也被广泛应用于各个领域。

计算机视觉系统在智能车辆的研究中主要起到环境检测和辨识的作用。为了简化视觉信息处理，通常会把智能小车的工作环境分为结构化道路环境和非结构化道路环境。结构化道路常用于研究阶段的算法验证，对结构化道路的检测相对来说比较容易实现，其检测技术一般都通过图像处理获取边缘信息，辅以 Hough 变换、道路模式匹配等方法，并采用最小二乘法处理对应道路边界的线条，得到道路的几何解。

减少图像处理计算量的基本思路是将图像做分割后再处理，一方面分割图像可以避免对无效像素的计算，将有限的计算资源用在感兴趣区域上，另一方面，对整幅图像处理的计算量要大于对分割后图像处理的总和，但分割处理后的信息拼接又会成为新的问题，另外，图像较小会引起目标与背景比例变化，可能造成信息提取错误。国内外学者为了提高路径识别算法的鲁棒性和实时性，提出了很多方法和独到的见解。王慧通过中值滤波和图像细化的方法，实现导航道路中心

线的提取和交叉线的去除^[31]。李进采用将路径图像分割上下两个区域分别计算路径信息的方法^[32]；汪明磊则将图像分割为四块，采用局部阈值的方法提取局部道路信息后合成整体信息，以加快运算速率^[33]。日本的 Masanori Hariyama 等将可靠地立体视觉匹配算法加载到一种 VLSI 处理器中，并设置参考图像，通过当前图像和参考图像间像素的迅速匹配，很好地解决了程序实时性的问题。法国的 Romuald Aufrere 提出了一种图像处理方法，利用单目视觉的车道模型匹配，可较精确地跟踪有路标情况下的道路。美国 MI 的 Y.LMurphey 等也通过单目视觉，测取图像中的深度信息，可较好地对目标进行定位。

1.3 本文研究目标

智能车作为移动机器人的一种，是具有自动驾驶能力的汽车。可以通过借鉴许多移动机器人的技术对它进行研究。路径跟随作为智能车辆系统的执行部分，是系统中不可缺少的重要组成部分，其跟随的准确性直接影响到智能车辆系统整体的检测决策。对已有路径的跟随算法研究，使得路径跟随也可作为一个单独的课题独立研究。智能车路径跟随技术类似于移动机器人路径跟随技术，但是对其车速以及行驶方向要求则更为严格，本课题主要是研究智能小车路径跟随问题。

本文以智能小车为主要研究对象，研究智能车的环境感知和路径跟随控制问题，包括：

（1）路径图像识别

采用单目视觉来获取图像信息，以图像处理的方式获取路径与智能车朝向的偏角大小。在模糊理论的基础上构造方向码盘，并通过隐式摄像机模型进行摄像机标定，获取变换矩阵后，将方向码盘做逆变换并与获取的路径图像信息比对，以得到与路径图像俯视图相同的偏角信息，减少整个路径识别算法的计算量，并通过理论和仿真实验验证其有效性。

（2）搭建智能小车模型

搭建 20:1 双轮差速驱动智能小车模型，以 STM32 开发板为智能小车控制板，设计外围传感器和控制电路。针对传感器和驱动电机，以及控制策略解决方案，开发编写控制程序，并采用开发工具提供的仿真器进行了初步的调试和验证。

（3）路径跟随控制

智能小车的路径跟随控制包括横向控制和纵向控制。本文采用模糊理论控制算法控制小车路径跟随，采用 PID 控制方法调节左右驱动轮转速实现小车按照模糊控制的输出指令行驶，并通过大量试验确定各个参数。控制方式采用多线程并行控制，以提高小车控制的稳定性和连续性。

1.4 本文组织结构

本文以图像识别和智能车的路径跟随底层控制为研究目标进行研究，各章的具体组织结构如下：

第1章中，介绍课题的研究背景与意义，智能车辆系统国内外的研究现状，以及本文的主要内容。并列举了一些路径识别算法。最后叙述了本文的研究工作、内容。

第2章中，首先对智能小车路径跟随系统作了需求分析，并确定采用远程闭环控制的方式，将整个系统分为上位机部分和移动端部分。建立两轮差速驱动小车的运动学模型。

第3章中，介绍本文涉及的有关图像处理的技术，和减少图像预处理计算量的算法改进。建立摄像机隐式模型并完成摄像机标定。

第4章中，详细说明了本文采用的模糊控制算法的思路，并通过仿真实验与传统的提取路径边缘线计算路径几何描述的预瞄控制算法的计算量和精确度做了比较，阐明了本文算法的优势。

第5章，搭建智能小车硬件平台并开发编程实现其控制，通过实验验证本文路径识别算法和智能车控制方法的正确性和可行性。

最后，对本文所做的工作和成果进行了总结，并指出了有待进一步研究的方面。

第2章 智能小车体系结构与建模

对于智能小车的嵌入式开发在人类的生产生活中并不少见。为了更加精确地控制小车，已有许多复杂的控制算法移植应用到智能车的控制上。但嵌入式控制板的性能有限，这导致一些像图像处理之类的过于复杂的算法无法在控制板中运行。而在研究阶段的智能小车的搭载能力有限，无法搭载性能较高的便携式计算机。为了解决这个问题，本文采用远程闭环控制的方法，将图像处理的程序算法放在上位机（PC）中运行，移动端小车作为行驶命令的接收和执行部分，与上位机相互配合，尽量减少车载嵌入式控制板的运算压力^[34]。

2.1 智能小车系统结构

智能小车系统集成环境感知、路径跟随等模块于一体，智能小车系统设计的目标是将车载传感器、跟随控制结构以及通信模块等功能有机地结合起来。智能小车的路径识别、路径跟随和控制算法在很大程度上由智能小车系统平台的硬件结构决定，因此，对智能小车结构的设计和运动学模型的建立为后续的工作提供了基础。

由于小车平台研究的最终目的是仿照驾驶员驾驶汽车行驶的过程，所以小车平台应从驾驶员的驾驶行为出发来构建。驾驶员的实际驾驶行为主要可分为三个环节：信息感知环节、轨迹决策环节以及操作环节，所以智能小车系统搭建也应基于这三个环节来考虑。

本文设计的主题是智能小车路径跟随系统的闭环控制，路径识别及图像处理的工作在上位机中完成，路径跟随及车速控制由下位机接收命令后在实验小车上执行。本系统包含硬件与软件两大系统，且每个系统又分为上位机与智能车两部分，下面分析各自应具备的功能。

上位机处理无线摄像头传回的图像信息，并将行进指令发送给智能小车，应具备的功能有：

- 1) 接入无线网络，接收局域网摄像头传回的图像；
- 2) 对接收到的图像作分析处理，计算前方路径相对当前小车当前朝向的偏差角；
- 3) 配置计算机串口，通过蓝牙串口主机与车载蓝牙串口从机通信，实现数据的发送与接收；

移动端负责智能小车的控制，应具备的功能有：

- 1) 车载局域网摄像头应接入网络，向上位机回传图像信息；
- 2) 解析串口接收到的控制命令，并根据命令驱动左右轮电机转动；
- 3) 检测左右轮转速并通过 PID 控制使车轮按照目标转速转动；
- 4) 通过串口向上位机回传小车左右轮速度信息。

根据系统需求分析，该系统由上位机与移动端车载控制板组成。上位机选择实验室内普通 PC 机，通过网线接入互联网，具有固定 IP，上位机使用 USB 转串口线连接无线蓝牙串口收发装置，移动端选择自行组装的智能小车，包括底盘、电机、电池、车轮，系统结构示意图如图 2.1 所示。

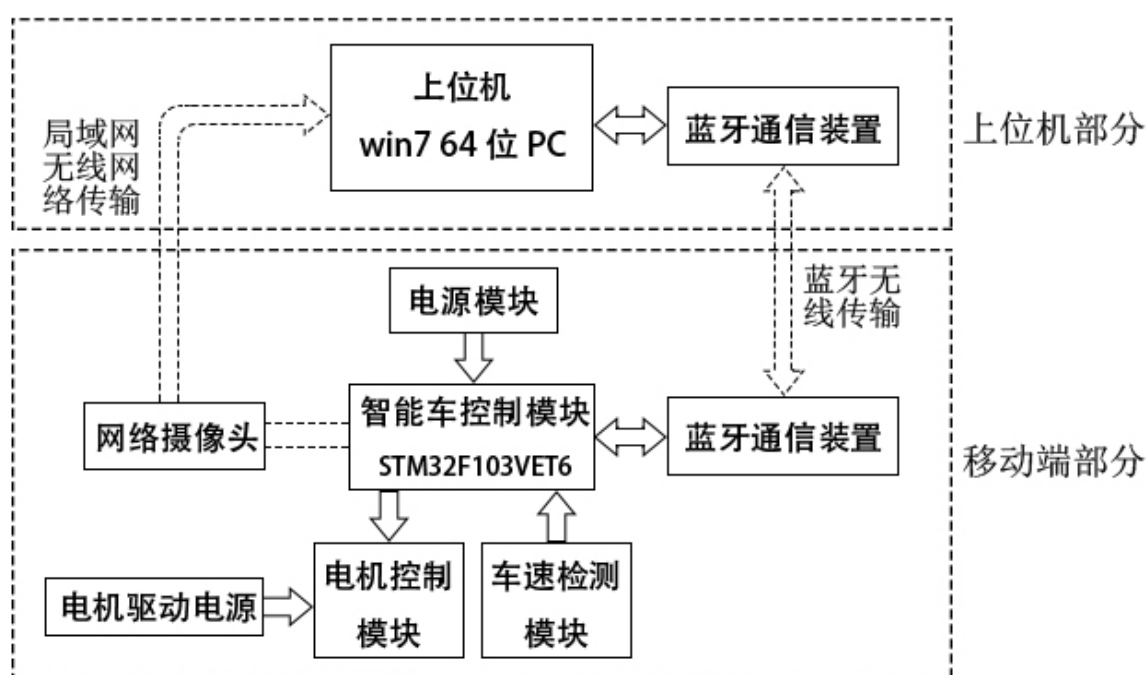


图 2.1 系统结构示意图

如图 2.1 所示，箭头方向代表控制信号和数据走向。系统分为上位机部分和移动端部分。移动端部分各个模块均搭载在小车底盘上。系统运行时，首先由网络摄像头将图像信息传送至上位机（PC）机，上位机中的控制程序完成图像处理分析后，将控制指令通过蓝牙串口通信装置的主机发送至移动端的蓝牙串口通信装置的从机，蓝牙通信装置将信号传至车载控制板。控制板将接收到的信号解码为具体的控制车轮驱动电机的 PWM 码，直接控制电机。车轮开始旋转后，车速检测模块将检测信号发送回车载控制板，轮速的计算过程在车载控制板内完成，计算出轮速后，控制板将速度信息传至蓝牙通信装置，发送至上位机，上位机接收小车左右轮旋转线速度后，可对小车运动中心的速度进行矢量积分获取小车位姿信息。整个系统成一个闭环结构。

2.2 智能小车运动学建模

本文选用智能小车底盘为双轮差速底盘，即没有转向舵机和转向轮，只有两个驱动轮，小车的转向通过两个驱动轮的差速旋转来实现。为了对智能小车系统的运动进行定量研究，必须对小车建立相符的数学模型，及建立系统的运动学模型。

2.2.1 智能小车的坐标系

物体的坐标系，可分为绝对坐标系和世界坐标系。智能小车的摄像头采集的图像信息和传感器探测的信息数据均是在智能小车的相对坐标系下的。智能小车在移动的过程中，若要确定小车的位姿信息，就需要在实验环境的绝对坐标系下对智能小车进行定位。定位的过程中，就需要对数据进行坐标变换。

智能小车相对坐标系如图 2.2 所示。坐标系原点定为两个驱动轮连线的中点。

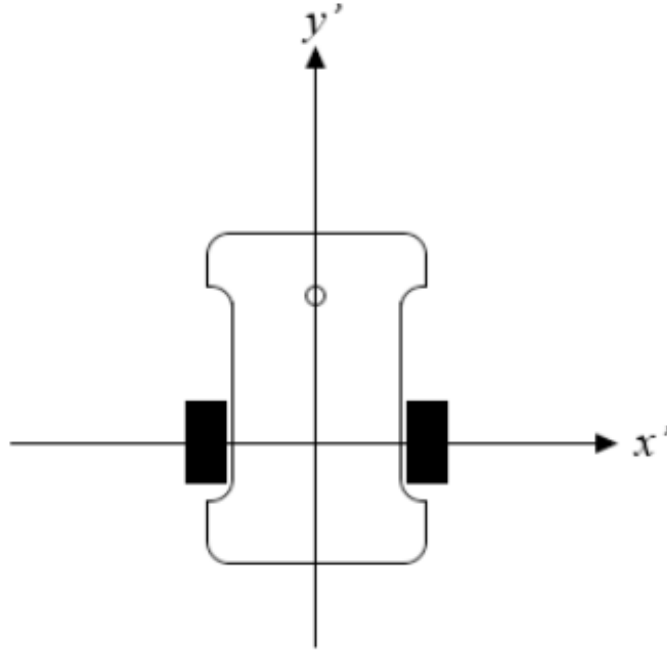


图 2.2 智能小车相对坐标系

双轮差速移动机器人在转弯时，需要两个驱动轮以不同速度行驶来实现，设左轮转速为 ωl ，右轮车速为 ωr ，车轮半径为 r ，假设在小车车轮在地面上做纯滚动无滑动的运动，则小车运动中心的移动速度 v 为：

$$v = \frac{(\omega l + \omega r)r}{2} \quad (2.1)$$

小车的转弯半径 R 为：

$$R = \frac{\omega l + \omega r}{\omega l - \omega r} \cdot \frac{L}{2} \quad (2.2)$$

其中, L 为小车两个驱动轮的轮距。

R 为正时, 表示小车向右侧转向行驶, R 为负时, 表示小车向左侧转向行驶。示意图如图 2.3 所示。

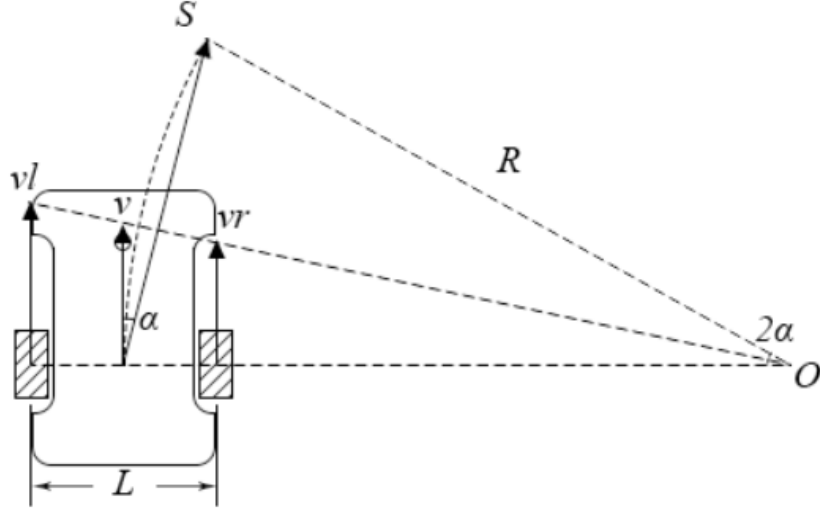


图 2.3 双轮差速转向示意图

小车在转向时, 实际上是在以 O 点为圆心, R 为半径做圆周运动, 考察时间 T 内, 小车实际走过的位移 \vec{k} 与相对坐标系 y' 轴的夹角为 α , 此时小车行驶方向偏离角为 2α , 则有以下式:

$$R \times 2\alpha = v \times T \quad (2.3)$$

联立式 (2.1)、式 (2.2) 和式 (2.3), 有:

$$\alpha = \frac{(\omega l - \omega r)Tr}{2L} \quad (2.4)$$

当 α 为正时, 表示小车向右侧转向行驶, R 为负时, 表示小车向左侧转向行驶, 为零时表示小车直行。

时间 T 内, 小车走过的位移为:

$$\vec{k} = [R - R \cos(2\alpha), R \sin(2\alpha)] \quad (2.5)$$

2.2.2 智能小车位姿描述

小车的行驶路径在绝对坐标系下描述, 在绝对坐标系中观察小车位姿, 才能验证路径跟随的效果, 因此需要对小车相对坐标下的位姿作坐标变换, 如图 2.4 所示。设小车初始位置与绝对坐标原点位置重合, s 为时间 T 内小车在世界坐标中走过的位移, θ 为小车朝向与绝对坐标系 y 轴的夹角, S 为小车在世界坐标下走过的总位移, 即小车在绝对坐标系中位置。小车的位姿可用 $[S, \theta]$ 来描述。

根据坐标变换, 有:

$$\begin{cases} \bar{s}_i = \bar{k}_i \cdot \mathbf{M} \\ \mathbf{M} = \begin{bmatrix} \cos(\theta_{i-1}) & \sin(\theta_{i-1}) \\ -\sin(\theta_{i-1}) & \cos(\theta_{i-1}) \end{bmatrix} \end{cases} \quad (2.6)$$

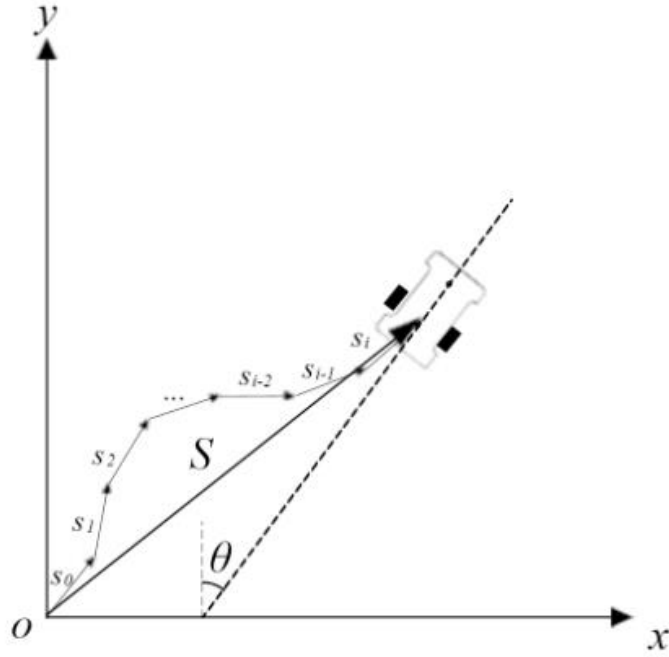


图 2.4 小车在绝对坐标系中的运动

其中， s_i 为当前周期小车在绝对坐标系下的位移； k_i 为当前周期小车在相对坐标系下的位移，由式(2.5)给出； \mathbf{M} 为从相对坐标系到绝对坐标系的旋转矩阵； θ_{i-1} 为当前周期开始时小车的朝向角，且有：

$$\theta_i = \sum_1^i 2\alpha_i \quad (2.7)$$

绝对坐标系下的位移 S ：

$$\bar{S}_i = \sum_1^i \bar{s}_i \quad (2.8)$$

2.3 本章小结

本章分析了智能小车的系统需求，根据需求设计智能小车系统结构，并介绍了各个组成部分的功能；建立了两轮差动驱动小车的运动学模型，为后续的控制以及数据分析奠定了基础。

第 3 章 图像处理与摄像机标定

摄像机获取图像信息后，并不能被计算机直接识别，需要对获取的图像做透视变换和目标提取。本章主要介绍图像处理设计的技术和摄像机标定的原理及过程。

3.1 图像处理基本原理

摄像机获取的图像为 RGB 彩色图像，且存在透视变形，因此需要对图像处理分析，消除噪音后提取路径信息。本节介绍图像处理涉及到的一些技术。

3.1.1 图像灰度化

RGB 彩色图像为三通道矩阵，图像的每个像素有 R、G、B 三个分量决定，而每个分量有 256 个可选值，若直接对 RGB 图像进行处理，耗时大，对系统整体的实时性影响较大。灰度图像只含有亮度信息，不包含色彩信息，将 RGB 彩色图像灰度化可大大减少计算量，提高运算速率。图像的灰度值实际上就是图像颜色的透明度，即亮度。式（3.1）为 RGB 到 YUV 的转换关系式^[44]：

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.144 \\ -0.148 & -0.289 & 0.437 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.1)$$

$$Y = 0.299 \times R + 0.587 \times G + 0.144 \times B \quad (3.2)$$

其中 Y 就是图像的亮度值， U 和 V 是色差信号。灰度图像为单通道图像，待处理数据为原来的 1/3。本文中将 RGB 图像转换为 256 个亮度等级的灰度图像，即灰度图像中矩阵元素取值范围为[0,255]。

3.1.2 最优阈值分割

阈值分割是按照一定的阈值将目标从图像的背景中分割出来，从而得到目标路径信息。阈值分割实际上是在对图像做二值化处理，分割过程，将图像中的每一个像素点与设定的阈值作对比，将高于阈值和低于阈值的像素划分开来，分为前景和背景，即完成了图像二值化的过程。阈值的设定直接决定了图像信息提取的准确率，因此，确定一个最优阈值非常重要。

在实验室的结构化路径实验中，智能小车采集的图像往往只包含两类主要的灰度值区域，即目标路径和背景，图像的直方图中，像素的灰度值概率分布密度

函数，即对应了目标路径和背景的两个单峰分布密度函数。如果目标和背景的密度函数是已知的，就可以计算最优阈值了，最优阈值一般取两个单峰密度函数间的谷值，如图 3.1 所示。在计算过程中，一般将目标路径和背景的密度函数近似认为是正态分布的，图像整体的概率密度可表示为：

$$p(z) = P_B p_B(z) + P_T p_T(z) = \frac{P_B}{\sqrt{2\pi}\sigma_B} \exp\left[-\frac{(z-\mu_B)^2}{2\sigma_B^2}\right] + \frac{P_T}{\sqrt{2\pi}\sigma_T} \exp\left[-\frac{(z-\mu_T)^2}{2\sigma_T^2}\right] \quad (3.3)$$

式 (3.3) 中， P_B 和 P_T 分别是背景和目标的先验概率，且满足 $P_B + P_T = 1$ ， μ_B 和 μ_T 分别为背景和目标的灰度值均值， σ_B 和 σ_T 分别是关于目标路径和背景的均方差。计算过程中需要求出一个阈值 T ，像素灰度值大于 T 的像素分割为目标而灰度值小于 T 的像素分割为背景，此时，将目标路径像素划分为背景区域的概率 $E_B(T)$ 和将背景像素划分为目标路径像素的概率 $E_T(T)$ 分别是：

$$E_B(T) = \int_{-\infty}^T p_T(z) dz \quad (3.4)$$

$$E_T(T) = \int_T^{\infty} p_B(z) dz \quad (3.5)$$

则总误分概率为：

$$E(T) = P_T E_B(T) + P_B E_T(T) \quad (3.6)$$

当满足 $\frac{dE(T)}{dT} = 0$ 时，可使 $E(T)$ 最小。得：

$$P_B p_B(z) = P_T p_T(z) \quad (3.7)$$

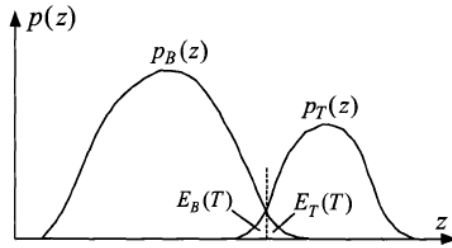


图 3.1 最优阈值确定示意图

综合可得：

$$\frac{(T - \mu_B)^2}{2\sigma_B^2} - \ln \frac{P_B \sigma_T}{P_T \sigma_B} = \frac{(T - \mu_T)^2}{2\sigma_T^2} \quad (3.8)$$

近似认为 $\sigma_B = \sigma_T$ ，若 $P_B = P_T$ ，则有：

$$T = \frac{\mu_B + \mu_T}{2} \quad (3.9)$$

由于 μ_B 和 μ_T 在实际中不易求出，最优阈值 T 的值也很难确定。为方便算法实现，先选取整体图像灰度的中值作为阈值 T 的初始值 T_0 ，然后通过迭代的方法求取其近似值。迭代步骤具体如下：

- 1) 计算整幅图像的最大灰度值 Z_{max} 和最小灰度值 Z_{min} ;
- 2) 设置初始阈值 $T_0 = \frac{Z_{max} + Z_{min}}{2}$, 和迭代时新旧阈值允许偏差 $\varepsilon = 0.5$;
- 3) 求出灰度值 $z_i \geq T_i$ 的像素点数量 n_T 与累积和 S_T , 以及灰度值 $z_i < T_i$ 的像素点数量 n_B 与累积和 S_B ;
- 4) 计算 $\mu_B = \frac{S_B}{n_B}$, $\mu_T = \frac{S_T}{n_T}$, 确定新阈值 $T_{i+1} = \frac{\mu_B + \mu_T}{2}$; 如果 $|T_{i+1} - T_i| \leq \varepsilon$, 则 T_{i+1} 为最终阈值, 否则, $T_i = T_{i+1}$, 转至步骤 3。

3.1.3 数学形态学操作

数学形态学的基本思想是用一定形状的结构元素来度量和提取图像中的形状, 以达到对图像进行分析和识别的目的。其基本运算有 4 个: 膨胀、腐蚀、开启和闭合^[43]。它们在二值图像和灰度图像中有各自的特点。

1) 膨胀

膨胀的算符为 \oplus , 用 **B** 来膨胀 **A** 写作 $\mathbf{A} \oplus \mathbf{B}$, 其定义为:

$$\mathbf{A} \oplus \mathbf{B} = \{x | [(\hat{\mathbf{B}})_x \cap \mathbf{A}] \neq \emptyset\} \quad (3.10)$$

上式表明用 **B** 膨胀 **A** 的过程是, 先对 **B** 做关于原点的映射, 得到如图 3.2(c) 所示的结构元素映像, 再将映像平移 x , 这里 **A** 与 **B** 映像的交集不为空集, 即用 **B** 来膨胀 **A** 得到的集合是 $\hat{\mathbf{B}}$ 的位移与 **A** 至少有一个非零元素相交时 **B** 的原点位置的集合。根据这个解释, 式 (3.10) 也可以写成:

$$\mathbf{A} \oplus \mathbf{B} = \{x | [(\hat{\mathbf{B}})_x \cap \mathbf{A}] \subseteq \mathbf{A}\} \quad (3.11)$$

式 (3.11) 可以帮助我们借助卷积概念来理解膨胀操作。如果将 **B** 看做 1 个卷积模板, 膨胀过程就是先对 **B** 做关于原点的映像, 再将该映像连续在 **A** 上移动实现的。

图 3.2 为膨胀运算的图解, 给出膨胀运算的一个示例, 其中图 3.2 (a) 中阴影部分为集合 **A**, 图 3.2 (b) 中阴影部分为结构元素 **B**, 图中标 “+” 处为原点, 它的映像如图 3.2 (c) 所示, 图 3.2 (d) 中两种阴影部分 (深色部分为扩大区域) 合起来为集合 $\mathbf{A} \oplus \mathbf{B}$ 。由图可见膨胀使图像区域扩大了一圈。

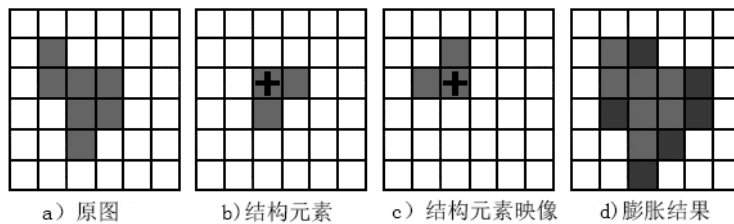


图 3.2 膨胀运算示例

2) 腐蚀

腐蚀的运算符为 \ominus ，用 **B** 来腐蚀 **A** 写作 $\mathbf{A} \ominus \mathbf{B}$ ，其定义为：

$$\mathbf{A} \ominus \mathbf{B} = \{x | (\mathbf{B})_x \subseteq \mathbf{A}\} \quad (3.12)$$

上式表明，**A** 用 **B** 腐蚀的结果是所有 x 的集合，其中 **B** 平移 x 后仍在 **A** 中。即用 **B** 来腐蚀 **A** 得到的集合是 **B** 完全包括在 **A** 中时 **B** 原点位置的集合。

腐蚀运算图解：

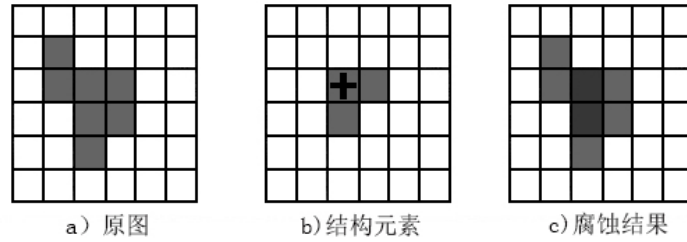


图 3.3 腐蚀运算示例

图 3.3 给出了腐蚀运算的 1 个示例。图 3.3 (a) 中的集合 **A** 与图 3.3 (b) 中显示的结构元素 **B** 都和图 3.2 中相同，而图 3.3 (c) 中深色部分表示 $\mathbf{A} \ominus \mathbf{B}$ 的结果，浅色区域为原属于 **A** 现被腐蚀掉的区域。由图 3.3 可见腐蚀运算将图像的区域缩小了一圈。

3) 开操作与闭操作

如上所述，膨胀运算使图像扩大一圈，腐蚀运算使图像缩小一圈，膨胀和腐蚀也因此得名。在实际操作中，经常将两种运算联合使用。先膨胀后腐蚀，或者腐蚀后膨胀，前者称为闭操作，后者称为开操作。开操作的作用是消除细的突出物和断开狭窄的间断；闭操作的作用与开操作相反，用以消除狭窄的间断和填补细小的孔洞，并连接轮廓线中的微小断裂。两者都可以使对象的轮廓线变得光滑。

3.2 二维平面的投影变换及标定方法

摄像机作为智能小车的眼睛，实现从三维景象到二维平面图像的转换。本文采用的摄像机标定模型为摄像机隐式模型^[38]，即不需要了解摄像机的详细物理参数，只需要知道景象是如何变换到图像这一变换关系即可。隐式模型可将摄像机的物理参数通过一定的变换，组合成一些中间参数，经过变换后，一般只需解线性方程组就可以标定摄像机，大大简化了标定过程。典型的隐式摄像机模型就是双平面摄像机模型，由魏国庆、马颂德在 1993 年提出^{[39][40]}的新双平面摄像机模型，将摄像机线性参数与非线性参数组合成一些中间参数，用线性计算方法标定。

本文的试验环境图像采集的景象为水平地面，网络摄像头固定于移动端智能小车上，故景象转为图像这一变换关系为固定不变，程线性关系，适合采用摄像机标定隐式模型。由于采集影像在水平地面的平面上，因此只需要二维坐标描述。

世界坐标系平面投影到摄像机像平面后，图像发生了变形，图 3.4 中 P_w 点投影在像平面上的点为 p ，像平面 UOV 坐标系与 $X_wO_wY_w$ 坐标系存在关系：

$$\begin{bmatrix} X_w \\ Y_w \\ 1 \end{bmatrix} = \mathbf{M} \cdot \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (3.13)$$

式中 $\mathbf{M} = \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & 1 \end{bmatrix}$ 为常数，是像平面坐标系转至世界坐标系的变换矩阵，

通过摄像机标定获得。

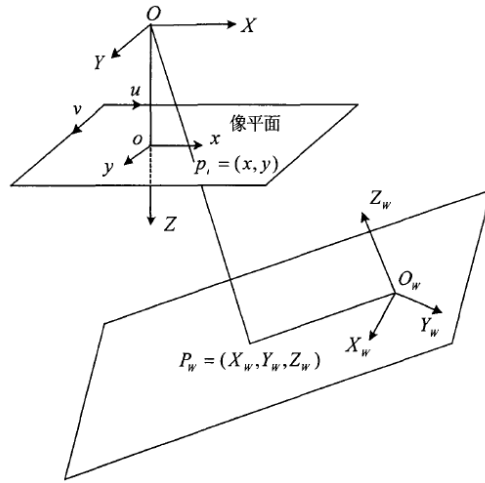


图 3.4 像平面与在摄像机三维空间中的二维平面之间的坐标关系

由于 \mathbf{M} 中有 8 个未知量，因此只需要在平面 $X_wO_wY_w$ 上标定 4 个非共线点，便可位移确定 \mathbf{M} 。且在标定这些点的时候，不需要考虑摄像机的内外部参数，因为这些参数已成为中间参数，隐藏在矩阵 \mathbf{M} 中。

设在平面 $X_wO_wY_w$ 上有 n ($n \geq 4$) 个点，则 $P_i = \{(X_i, Y_i), i=1, 2, \dots, n\}$ ，与其对应的像素点 $I = \{(u_i, v_i), i=1, 2, \dots, n\}$ 。

根据式 (3.13)，可得到一系列方程如式 (3.14)。

$$\begin{bmatrix} u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1 X_1 & -v_1 X_1 \\ 0 & 0 & 0 & u_1 & v_1 & 1 & -u_1 Y_1 & -v_1 Y_1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_n & v_n & 1 & 0 & 0 & 0 & -u_n X_n & -v_n X_n \\ 0 & 0 & 0 & u_n & v_n & 1 & -u_n Y_n & -v_n Y_n \end{bmatrix} \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \end{bmatrix} = \begin{bmatrix} X_1 \\ Y_1 \\ \cdot \\ \cdot \\ \cdot \\ X_n \\ Y_n \end{bmatrix} \quad (3.14)$$

式 (3.14) 可写成： $\mathbf{Hm}=\mathbf{N}$ 。在实际应用中，可使用多元线性回归参数的估计法^[45]对这种线性系统求解。

实际标定时，在平坦地面上放置 5 个尺寸为 $5 \times 5\text{cm}$ 的标定模块，用安装在智能小车上的摄像头对拍摄标定模块进行拍摄，如图 3.5 所示。摄像机标定时的安装位置需要保持与路径跟踪时的位置相同，这样才能保证 \mathbf{M} 的有效性。表 3.1 记录了 16 个标定点的坐标值。

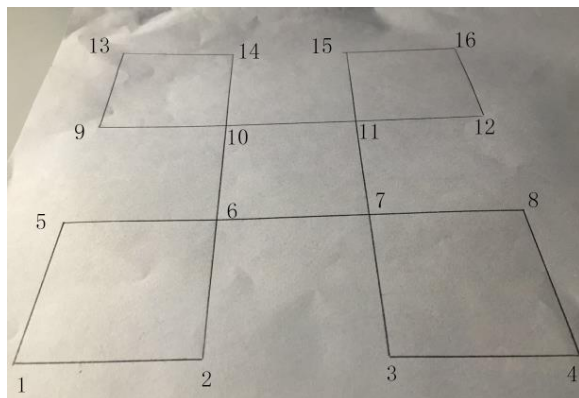


图 3.5 标定模块示意图

表 3.1 标定点点坐标值

序号	$X_i(\text{cm})$	$Y_i(\text{cm})$	u_i	v_i	序号	$X_i(\text{cm})$	$Y_i(\text{cm})$	u_i	v_i
1	0.0	3.0	306	35	9	0.0	13.0	260	160
2	5.0	3.0	206	37	10	5.0	13.0	193	161
3	10.0	3.0	106	38	11	10.0	13.0	124	163
4	15.0	3.0	7	40	12	15.0	13.0	57	166
5	0.0	8.0	279	110	13	0.0	18.0	247	199
6	5.0	8.0	198	111	14	5.0	18.0	190	197
7	10.0	8.0	117	114	15	10.0	18.0	129	199
8	15.0	8.0	36	117	16	15.0	18.0	72	202

通过回归算法^[45]，可计算出投影变换矩阵

$$\mathbf{M} = \begin{bmatrix} -0.045799 & -0.016476 & 14.591103 \\ 0.000694 & 0.041920 & 1.040635 \\ -4.091894 & -0.002307 & 1 \end{bmatrix} \quad (3.15)$$

第 4 章 路径特征提取与路径跟随控制

基于机器视觉的路径跟随智能车辆，其关键技术是计算机视觉技术和控制技术。在完成路面环境信息的获取后，需要通过路径识别和智能控制使智能车辆沿目标路径行走。国内外在智能车辆的控制方面已做了大量的研究工作^[46-47]，取得了良好的效果，但仍然存在不少问题，如控制的实时性和鲁棒性。

考虑到驾驶员的驾驶行为：驾驶员在开车时，总是不断地对前方的路径信息进行预瞄，然后再根据前方路径的弯曲程度来决定车辆的转弯方向、转角大小以及车速大小。基于图像识别的智能车辆的路径跟随系统，其驾驶员模型应该与汽车驾驶员的驾驶行为相似。

4.1 方向码盘

目前的路径识别的方法，多为将图像信息处理后提取道路中心线，采用预瞄的方法计算道路中心线的曲率，通过计算的曲率来控制智能车沿目标路径行走。计算曲率可以精确地获得目标路径的信息，但计算量较大，每个周期都需要对图像做逆透视变换并提取路径边缘线，且依赖于精准的车体运动学模型。因此，本文提出了一种基于模糊控制的路径特征提取的方法，采用方向码盘与目标路径作对比，以获取路径走向信息。

4.1.1 方向码盘概念的提出

想象人在驾驶车辆对一条地面标识路径跟随时，会注视前方地面路径，观察路径的走向，如图 4.1（a）所示。但人在观察的过程中，并不会求取路径中心线的几何描述，而只是有一个感官上的认识，即路径走向与车辆当前朝向是否存在较大偏差角。若偏角较小（深色区域范围），如 A 线路所示，则会做出“略微转向”的决策，若偏角较大（浅色区域范围），如 B 线路所示，则会做出“转大弯”的决策，整个比较过程都在相对坐标系下完成。当预瞄距离确定时，对路径走向和车辆朝向偏角大小的判断，亦可等价为路径和相应的圆弧是否有交叉的问题，如图 4.1（b）所示。当把这种思路应用于计算机图像处理时，判断圆弧与路径是否有交叉比计算相应的偏角大小更便捷。将 4.1（b）的预瞄圆弧扩展到车辆行驶方向的整个正前方，且考虑计算机图像处理到车辆行驶指令的时间差以及一些其他行驶因素，将方向码盘的起始线向车辆正前方推移，如图 4.1（c）所示。

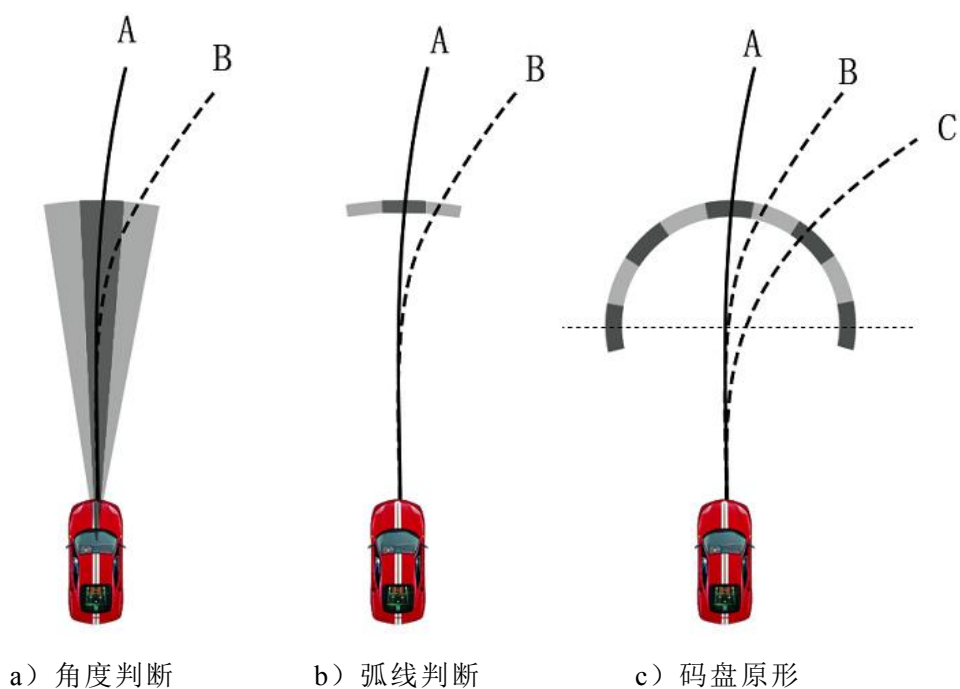


图 4.1 方向码盘概念提出

当驾驶员在驾驶车辆行驶时，若车速较高时，会将更多注意力集中在车辆前方较远处，即预瞄距离较大，若前方出现急弯，又会将视线收回近处。因此本文设置两个虚拟方向码盘，分别为 NEAR 码盘和 FAR 码盘，两个码盘的中间区域为过渡区域。

以小车正前方方向为中心线， 30° 角为划分单元，将车辆前行方向划分为 7 个区域，如图 4.2 所示。

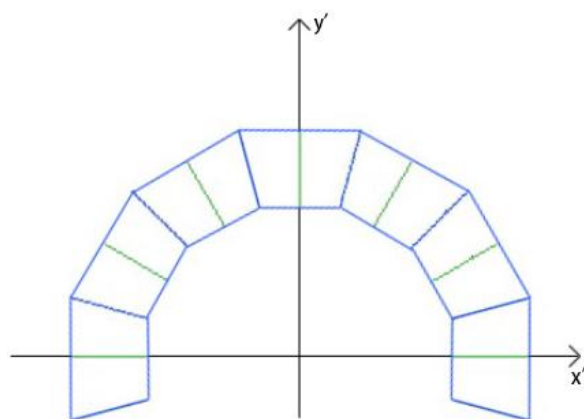


图 4.2 方向码盘

4.1.2 方向码盘透视变化

在车辆行驶过程中，方向码盘固结于小车的相对坐标系上，即在车辆前行过程中，在地面虚拟一个方向码盘，如图 4.3 所示。摄像头采集的图像为透视图，需要将路径图像做你透视变化后才能与方向码盘比对。逆透视变化关系由式(3.15)给出。

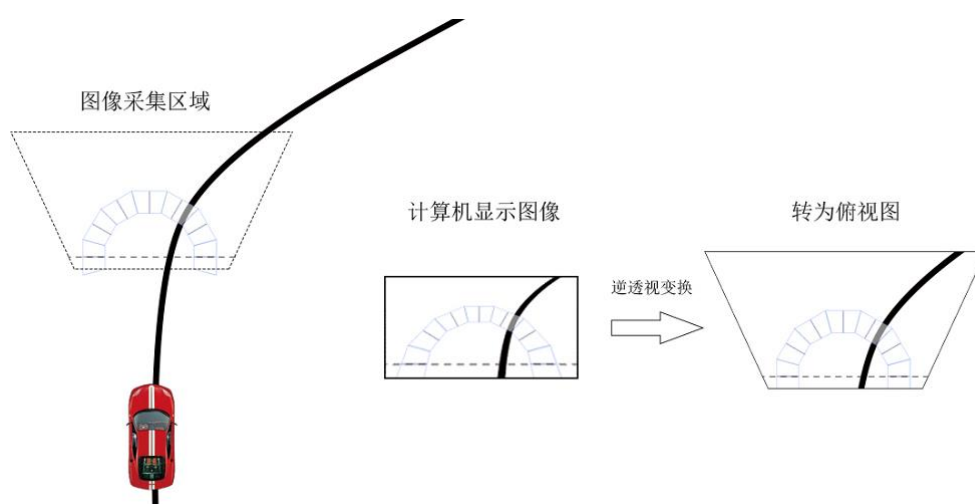


图 4.3 虚拟方向码盘使用

图 4.3 中的图像采集区域和计算机显示图像实际上并无方向码盘，图示为了清楚对比路径图像信息，将虚拟方向码盘加上。由图 4.3 可知，在透视图与俯视图的变换过程中，方向码盘与路径图像信息的交叉关系保持不变。在实验中，路径信息处于变化中，而方向码盘的形状固定不变，出于减少计算量考虑，可通过将经过透视变形的方向码盘直接与计算机显示图像对比的方式，避免每次获取路径图像后都需要做逆透视变换。根据式 (3.13)，对方向码盘做透视变换， \mathbf{M} 由式 (3.15) 给出，方向码盘透视变换如图 4.4 所示。

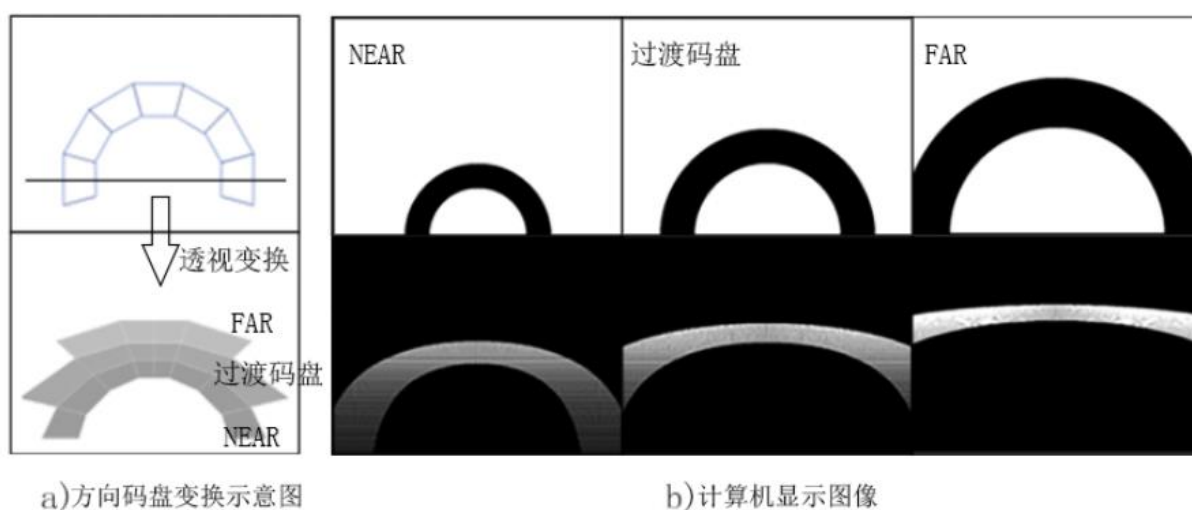


图 4.4 方向码盘透视变换

4.2 模糊控制器的设计

模糊控制是模仿人类决策时的模糊判断，运用模糊数学的基本思想和理论的控制方法，属于智能控制的范畴。驾驶员在驾驶汽车时，如遇前方车道笔直且畅通时会适当提速；在看到前方行驶路径出现弯道时会提前减速准备过弯。实验小车控制策略模仿人的驾驶习惯，在此基础上参照模糊控制的输入量定性测量的方

法，在满足目标的前提下减少计算量，实现对目标路径的快速跟随。

4.2.1 输入量与输出量的确定

控制器的输入是目标路径与小车当前朝向的偏角大小、小车左右轮运动速度。输出是控制小车左右轮转速、小车行驶指令及小车预瞄距离 D 。传至上位机的图像中包含大量视觉信息，经过图像处理将路径信息提取后，与方位码盘作对比，得出路径走向。将路径走向 θ 输入变量的论域划分为 $\{LB, LM, LS, FRONT, RS, RM, RB\}$ ，将当前运动速率 v 的论域划分为 $\{SLOW, FAST\}$ ；将实验小车前行方向 β 论域划分 $\{NB, NM, NS, Z, PS, PM, PB\}$ ，将预瞄距离 D 论域划分为 $\{NEAR, FAR\}$ 。

对输入量设定隶属度函数，出于减少计算量考虑，取各语言变量隶属度函数的形状为对称的三角形。规定路径走向 θ 左为负反之为正。路径走向 θ 的隶属度函数如图 4.5 所示，当前运动速率 v 的隶属度函数如图 4.6 所示。

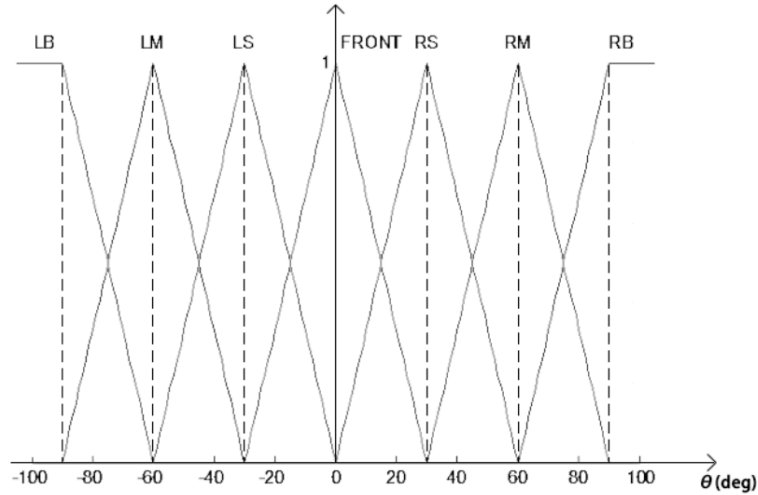


图 4.5 路径走向 θ 的隶属度函数

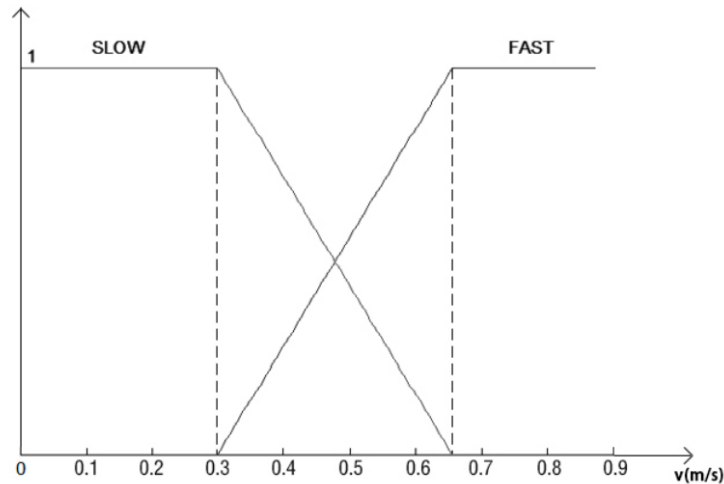


图 4.6 小车当前速率 v 的隶属度函数

4.2.2 模糊规则

控制策略是控制算法的核心部分，主要在上位机中完成计算，并将控制命令发送至车载控制板上。实验小车在路径跟随过程中，若出现持续向正前方行驶，则将预瞄距离 D 拉长，同时提高小车行进速率。在预瞄距离 D 置 FAR 档时检测到大弯道则将 D 置 NEAR，同时降低小车速率，准备过弯。过弯过程中根据路径与方向码盘的对照得出路径走向 θ 的值，根据路径走向 θ 输出下一检测周期小车的前行方向 β 。对于输出量预瞄距离 D 及小车前行方向 β ，制定模糊规则，见表 4.1。表 4.1 里每格里两个输出量，分别为小车前行方向 β 及预瞄距离 D 。

表 4.1 小车前行方向 β 及预瞄距离 D 模糊推理规则

$\begin{matrix} v \\ \theta \end{matrix}$	SLOW		FAST	
LB	NB	NEAR	Z	NEAR
LM	NM	NEAR	Z	NEAR
LS	NS	NEAR	NS	NEAR
FRONT	Z	FAR	Z	FAR
RS	PS	NEAR	PS	NEAR
RM	PM	NEAR	Z	NEAR
RB	PB	NEAR	Z	NEAR

表 4.1 中每组两个输出量分别为小车前行方向 β 及预瞄距离 D ，预瞄距离 D 的论域 NEAR 和 FAR 除了控制预瞄距离外，还对应了小车接下来的运动中心速度 v ，NEAR 对应慢速，FAR 对应快速。小车运动中心速度 v 决定后，再根据 β 确定小车的行驶圆弧的半径 R ，Z 为直行， β 为 NS 与 PS 时，对应的小车行驶的圆弧路径半径较大，即转向角较小，NM 与 PM 对应的小车行驶的圆弧路径半径较小，即转向角较大，NM 与 PB 对应的行驶路径圆弧半径最小，小车转向角最大，具体计算方式见本章 4.3 节。

智能小车在行驶中会有提速和减速的过程。当车速 v 为 SLOW 时，若此时检测到路径方向为 FRONT，则输出的前行方向 β 和预瞄距离 D 分别为 Z 和 FAR，如表 4.1 所示，但此时小车并不会立即提速，而是开启提速计数器，并继续按照低速状态行驶，当输出量预瞄距离 D 连续 k 周期为 FAR 时，则开始提速， k 为预先设定的值。反之，当 v 为 FAST 时，若输出量预瞄距离 D 为 NEAR，则立即减速。

上位机每个周期 T 从网络摄像头传回的图像信息中截取一幅图像，将图像二值化并通过开操作消除与路径图像相连的较小面积的连通域，而后与方位码盘作对比，图像处理过程如图 4.5 所示。

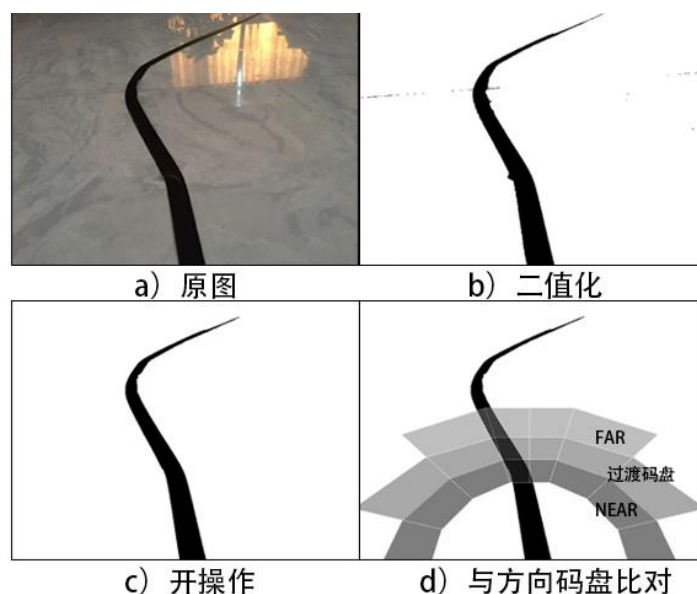


图 4.5 图像二值化与方向码盘示意图

对路径信息与方位码盘重叠部分的连通域作检测，并检测连通域重心位置。记方位码盘中线左侧为负，方位码盘从 -90° ~ 90° 共划分为 7 块区域，分别对应路径走向 θ 的论域划分。 θ 所在论域即为重心位置所在区域的对应论域。若连通域重心位置落在区域划分线上，则取大值，即按照 FRONT, S(small), M(middle), B(big) 的顺序优先取值。

4.3 小车运动控制

小车在前进过程中，当检测到前方路径出现弯道时，会接收到转弯行驶指令。由于本文采用的是指令延时发送的方式，指令何时发送、小车在何时接受指令并如何执行以到达目标点是本文要考虑的关键问题，也是小车能否准确沿着路径行走的关键。

4.3.1 智能小车指令执行时间

现分析小车接收指令并执行的两种时刻及其效果。

(1) 当小车行驶到方向码盘起始线时执行指令

小车在行驶到方位码盘起始线时开始执行转向指令，对小车运动分析，如图 4.6 所示。假设检测到的路径走向在论域 RS 中，则小车执行行驶指令的最终目标是行驶到此时标出的灰色区域中。双轮差动小车在过弯行驶时，实际上是在走圆弧，以圆点表示小车运动中心，则小车从方位码盘行驶到图 4.6 所示 A 点位置的路径在图中用虚线圆弧绘出，小车围绕圆心 o' 做圆弧运动。小车到达 A 位置后车身朝向如图 4.6 的 A 点 v 箭头方向所示。

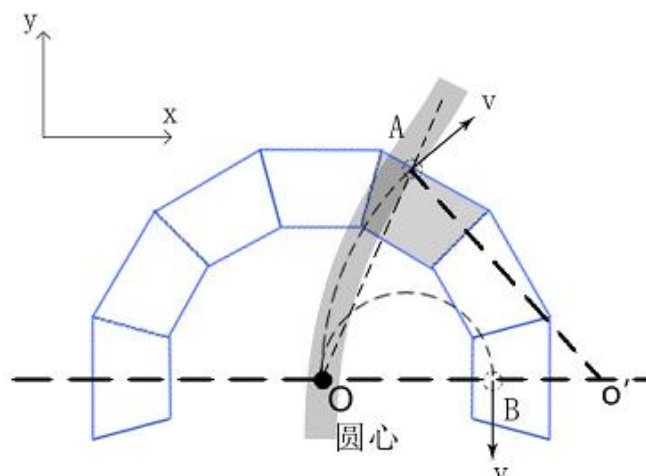


图 4.6 行车指令执行情况 1

不失一般性地考虑，当路径的图像落在论域 RS 中时，则认为路径接下来的方向也在论域 RS 的范围内。则当小车行驶到位置 A 处时，若小车此时的朝向与 OA 方向重合，则更利于接下来的图像采集与小车控制。根据第二章运动学模型所述，小车到达 A 位置时，车身朝向 v 与 y 轴的夹角为 $\angle AOB$ 的两倍。

在考虑特殊情况如路径图像落在 RB 中，如图中 B 点所示。根据小车的圆弧行驶路径，小车从 O 点到 B 的行驶路径如图中圆弧 OB 所示，此时小车朝向如 B 点 v 所示，而此时的期望朝向为 OB 方向，则实际朝向与期望朝向成直角，若此时需要调整朝向，则又会重复从 O 到 B 的行驶过程，出现不稳定的情况，且图像信息的采集过程也会受到影响，甚至直接偏离路径，因此，小车从 O 点开始接收并执行指令不利于小车的稳定行驶。

(2) 小车在行驶到方向码盘起始线前执行行驶指令。

同样以圆点表示小车运动中心，绘制小车行驶路径示意图，如图 4.7 所示。

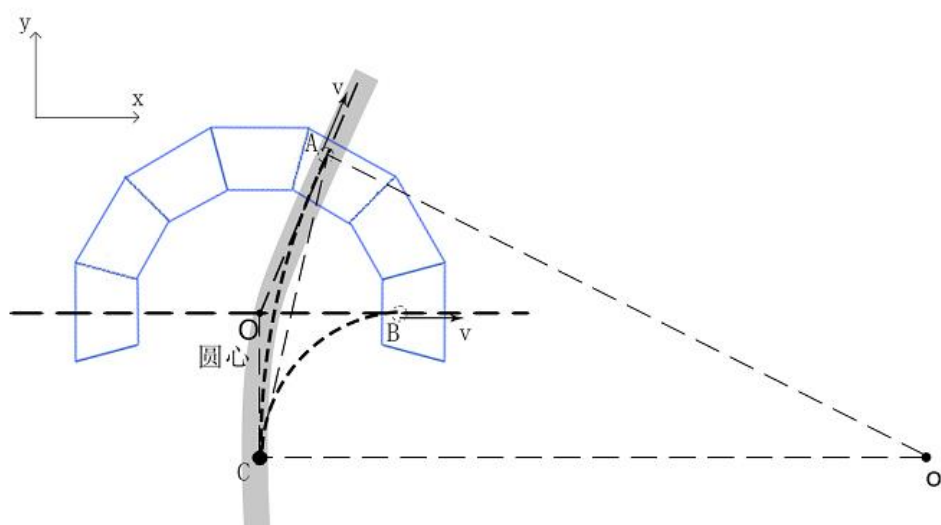


图 4.7 行车指令执行情况 2

设小车在 C 点接收到指令并执行, 目标点为 A 点, 小车初始朝向与 y 轴正方

向重合，小车到 A 点的行驶路径如圆弧 CA 所示，运动圆心为 o' ，到达 A 点后小车朝向如 A 点 v 所示，此时小车朝向与 OA 方向较为接近，如果取 $OC=OA$ ，则小车到达 A 点朝向与 OA 方向重合。若目标点为 B 点，按上述方式行驶，到达 B 点朝向与 OB 方向重合。此方法可行。

4.3.2 小车左右轮速度

确定小车指令执行时间后，接下来要解决发送怎样的指令使小车在该情况下到达 A 点，即确定小车左右轮的转动线速度。如图 4.7 所示，连接 Oo' ， $\angle o'OC$ 的大小为 $\angle AOC$ 的一半，根据第二章所述，当出现路径没有落在论域 FRONT 而是其它论域需要转弯行驶时，则按最小的偏航角来行驶，例如，当路径落在论域 RS 中时，则将目标点 A 点置于论域 FRONT 与 RS 的交界线上，即 $\angle AOB=75^\circ$ ，则小车运动轨迹圆弧的半径为：

$$R = R_M \tan(90 - \alpha / 2) \quad (4.1)$$

其中 R_M 为当前方向码盘的平均半径， α 为 4.2 中所述的不同论域对应的最小偏航角。

求得运动轨迹圆弧半径后，即可根据式(2.4)变形可得：

$$vl - vr = \frac{L}{2R} \cdot (vl + vr) \quad (4.2)$$

在小车行驶过程中，控制小车运动运动中心速度在同一论域中保持不变，则 $(vl + vr)$ 为已知项，则可根据式（4.2）求解得到：

$$\begin{cases} vl = v(1 + \frac{L}{2R_M} \cdot \tan(\frac{\alpha}{2})) \\ vr = v(1 - \frac{L}{2R_M} \cdot \tan(\frac{\alpha}{2})) \end{cases} \quad (4.3)$$

由式（4.3）可以计算式（4.2）中模糊规则输出的各个指令对应的左右轮旋转线速度。

4.3.3 小车偏航复位

小车在行驶过程中，可能会出现与路径线偏离并与路径线平行的情况，如图 4.8 所示。

此时若输出转弯指令，会使小车在路径跟随过程中出现方向波动，理想行驶路线如虚线所示。为避免这种情况对路径跟随过程的干扰，在图像预处理过程中，加入起始线对中功能，路径图像对中后，与方向码盘比对结果为 FRONT，记录对中位移 px ，记位移方向向左为正，向右为负，设定阈值 k ，当 $|px| < k$ 时，则小车继续按原行驶指令行驶，当 $px > k$ 或 $px < -k$ 时，则在原行驶指令基础上叠加偏航复位速度，即：

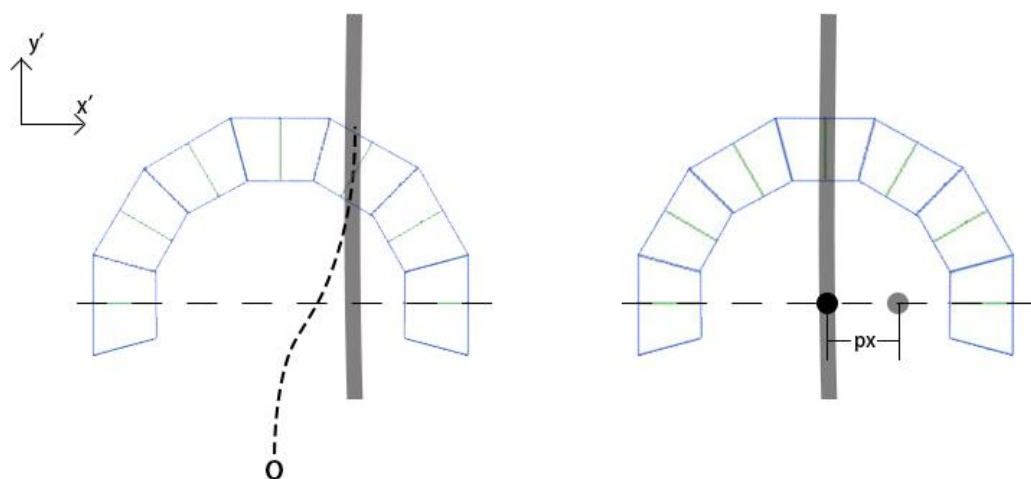


图 4.8 小车偏航复位

$$\begin{cases} vl = vl' + vp \\ vr = vr' - vp \end{cases} \quad (4.4)$$

其中 vl' 和 vr' 为模糊控制输出的左右轮行驶速度， vp 为偏航复位速度，有：

$$vp = \begin{cases} v/20, & px > k \\ -v/20, & px < -k \end{cases} \quad (4.5)$$

通过对 k 值得设定，即可控制小车在路径跟随的过程中与目标路径的允许偏差，经过实验仿真，确定 k 值为 $L/10$ ， L 为小车的左右轮的轮距。

4.4 模糊控制规则的补充

根据 4.3.2 的讨论，可确定各个输出论域对应的左右轮转速。以低速状态为例，设置小车运动中心速率 $v=0.4\text{m/s}$ ，则各个论域对应的左右轮转速见表 4.2。

表 4.2 输出指令与对应左右轮速度

小车前行方向	vl (m/s)	vr (m/s)
NB	0.00	0.80
NM	0.20	0.60
NS	0.34	0.46
Z	0.40	0.40
PS	0.46	0.34
PM	0.60	0.20
PB	0.80	0.00

由表 4.2 可知，在 NS 与 PS 之间，输出的行驶指令均为向正前方行驶，但在实际方向码盘的划分中，FRONT 对应的偏向角度为从 -15° 到 15° 的 30° 角，在方

向码盘与小车距离较近时， 30° 角的范围覆盖了较大的行驶区域，一直保持直行会让小车与目标路径的偏差较大，且容易很快进入 LS 与 RS 这两个相邻区域，引起较大的角度修正，不利于横向操纵的稳定性。因此，需要在输出指令为 Z 的情况下进一步细分指令，以实现小车在 Z 指令下的行驶方向微调，提高行驶的平顺性。

Z 指令对应的 FRONT 区域在小车当前朝向的正前方，可将弧度近似为直线段，将直线段等分为 7 段，方向码盘半径为 R_M ，包括该段圆弧的两个端点共 8 个点，设为

$$Pt(i) = -R_M \sin(15^\circ) + \frac{i-1}{7} \cdot 2R_M \sin(15^\circ), i = 1, 2, \dots, 8 \quad (4.6)$$

在图像处理中求出路径图像与方向码盘重叠部分的重心坐标 (C_x, C_y) ，该坐标在小车的相对坐标系下描述，根据 C_x 的值输出小车左右轮的轮速，见表 4.3。

表 4.3 在 Z 指令下的行驶方向微调

C_x	vl (m/s)	vr (m/s)
[Pt(1),Pt(2))	0.36	0.44
[Pt(2),Pt(3))	0.38	0.42
[Pt(3),Pt(4))	0.39	0.41
[Pt(4),Pt(5)]	0.40	0.40
(Pt(5),Pt(6)]	0.41	0.39
(Pt(6),Pt(7)]	0.42	0.38
(Pt(7),Pt(8)]	0.44	0.36

由于采用模糊控制，屏蔽了精确的小车运动学模型，在路径跟随的过程中，会牺牲一部分的跟随精确度，但在同时可以提高路径跟随的实时性。4.3 节中小车偏航复位的过程亦是模糊控制规则的一个补充。

4.5 上位机（PC）控制软件功能设计

对智能车系统做需求分析，上位机要实现的功能主要有：

- 1) 获取网络摄像头传回的图像信息；
- 2) 对获取的图像信息作分析计算；
- 3) 通过蓝牙串口向小车发送行驶指令；
- 4) 通过蓝牙串口接收小车回传的速度信息。

根据需求分析，将上位机程序模块化，主要分为两部分，分别为图像处理、串口数据收发，上位机程序处理过程如图 4.9 所示。上位机程序设计为多线程程

序，主线程负责计算行驶指令，副线程负责计时并向小车发送行驶指令。当读取不到路径信息时，会按照上一指令行驶 3 个周期，若还未检测到路径信息，即发送停止指令，这样设计可以使小车具备越过短距离的路径破损区域。

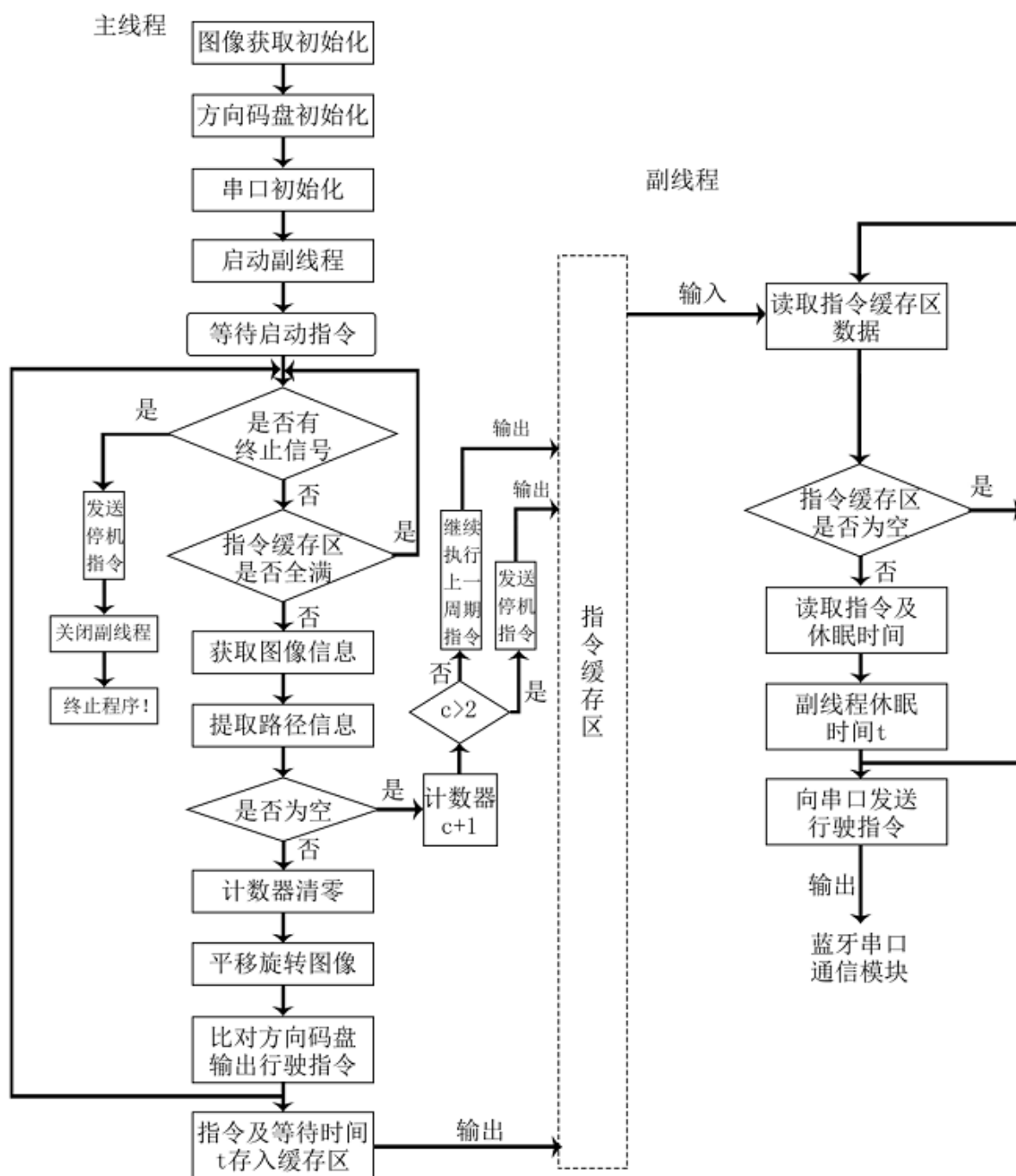


图 4.9 上位机程序流程图

4.5.1 上位机（PC）软件开发环境

在开发环境的选择时，考虑到所选 PC 机操作系统为 Windows7，因此选用微软公司推出的 Microsoft Visual Studio 2010。Visual Studio 是微软公司推出的开发环境，是目前最流行的 Windows 平台应用程序开发环境。Visual Studio 2010 版本

于 2010 年 4 月 12 日上市，支持开发面向 Windows7 的影响程序。

4.5.2 图像接收处理的实现

本文图像处理主要借助 OpenCV 的图像类 Mat 及相关图像处理函数。OpenCV 的全称是：Open Source Computer Vision Library，是一个基于 BSD 许可发行的跨平台计算机开源视觉库，可以运行在 Linux、Windows 和 Mac OS 操作系统上。它轻量级而且高效，由一系列 C 函数和 C++ 类构成。本文使用 OpenCV 库版本为 2012 年发布的 2.4 版本。Mat 即矩阵 (Matrix) 的缩写，是 C++ 类，也是 2.4 版本 OpenCV 中最基本的数据结构。

根据第三章所述，无线摄像头传回的图像信息显示在浏览器页面中，因此首先需要捕获图像信息，本文采用的方法是直接读取显示器内存中的数据，将数据存入 Mat 类的实例中，具体方法为在主程序 main() 中调用 InitPrscrn() 函数及 CopyToGray()。

主要代码如下：

```
InitPrscrn()
{ ...
hSrcDC = CreateDC(L"DISPLAY",NULL,NULL,NULL); //创建显存句柄
hMemDC = CreateCompatibleDC(hSrcDC);
//创建一个与显存兼容的内存设备上下文环境
hBitmap = CreateCompatibleBitmap(hSrcDC, nWidth, nHeight);
//创建指定尺寸的与显存兼容的位图
}
CopyToGray()
{...
BitBlt(hMemDC, 0, 0, nWidth, nHeight, hSrcDC, Left, Top, SRCCOPY);
//将指定区域的像素进行位块转换，并传入hMemDC中
...
pBuffer = new BYTE[nWidth*nHeight*image_nchannels]; //创建缓冲区
cv::Mat a ( nHeight,nWidth,CV_8UC4); //确定容器大小
GetBitmapBits(hBitmap,nHeight*nWidth*image_nchannels,pBuffer);
//将位图存入缓冲区
memcpy(a.data,pBuffer,nHeight*nWidth*image_nchannels);
//将缓冲区的数据存入a中
cv::cvtColor(abc,abc,CV_RGBA2GRAY); //图像灰度化
return a; //返回图像
```

```
}
```

InitPrscrn()函数的作用是初始化截屏功能,设定从显存中读取数据的范围(即截图的尺寸)。

CopyToGray()函数的作用是新建用于接收数据的Mat类实例a,并将读取的数据存储到该实例a中,此时a中存储的图像信息便是用户指定范围内的彩色图像信息,在该图像做灰度处理后,返回该图像。

从显示器获取图像信息后,主函数main()主要调用四个函数对获取的图像信息做处理,分别为blackline()、MoveToCenter()、rotatem()、findcenter_x()、getorder()。

Blackline()函数的作用为将图像信息二值化,并通过腐蚀、膨胀等方法消除道路线以外的干扰信息。

MoveToCenter()函数的作用为将blackline()返回图像的起点位置平移到考察矩形范围的中间位置,避免因道路线少量的横向偏移而错误地给小车发送转向命令。

findcenter()函数的作用是,求出道路线与方向码盘的重合区域,并求出重合区域的质心坐标。

getorder()函数的作用是根据findcenter()函数返回的坐标值,按照事先约定的区域划分,计算出应当给小车发送的行驶指令。

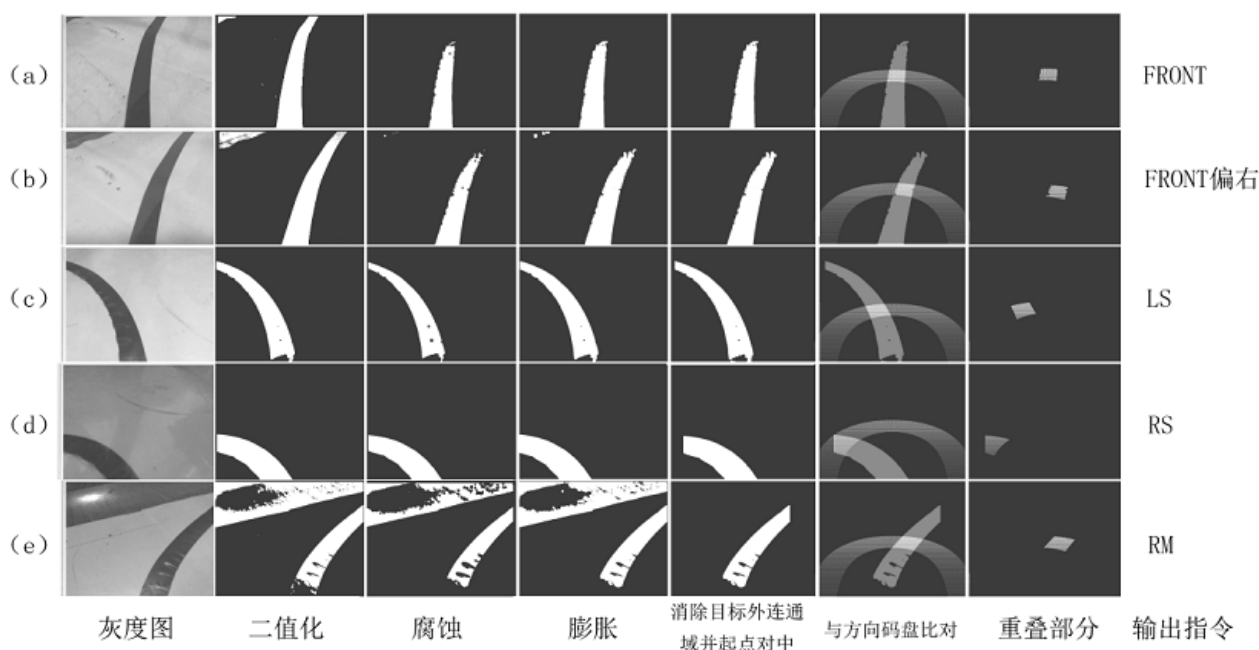


图 4.10 图像处理过程

图像处理过程如图4.10所示,从左到右依次为:二值化、腐蚀、膨胀、消除路径以外的大面积连通域并起点对中、与方向码盘比对、提取重叠部分、根据重叠部分重心位置所在论域输出指令。

4.5.3 串口数据收发

主程序中调用的串口相关函数主要有 InitComm()、CommWrite()、CommRead()，并通过以下代码将从串口读取的数据储存：

```
str2=Comm1.CommRead();    //从串口一次读入数据
while(Comm1.ReadBytes());//等待串口数据读取完毕
fout.open("rec.txt",ios::app);//以在内容后添加的写入方式打开文本文件
fout<<str2<<std::endl;//将读取到的数据写入文本文件中，并换行
```

InitComm()函数的主要功能是打开并初始化串口，将串口配置为与移动端串口模块相同的 115200-8-1-N（即波特率 115200，一次传输 8 字节，停止位为 1，无校验符），只有配置完全相同，才能建立通信。

CommWrite()函数的作用是将数据发送至移动端串口通信模块；CommRead()函数的作用是从上位机串口读取数据。为不影响主线程的图像处理，这两个函数均在副线程中调用。

4.6 实验仿真

为验证本文算法的高效性，在 VS2010 环境下编写仿真程序，与传统的预瞄控制算法作对比。

跟随的路径以图像形式输入，路径宽度为 7.5mm，仿真实验车速设定为 0.4m/s，图像采集周期为 0.1s，预瞄距离 d 为 6cm，小车两驱动轮间距为 150mm，实验输出为小车两驱动轮的转动线速度。

传统路径识别方法是先通过 Canny 算子提取道路边缘线，然后选择计算量较小的最小二乘法提取路径信息。仿真结果如图 4.11 所示。

由图 4.11 可知，在路径跟随的准确性上，曲线道路时传统的预瞄控制算法更有优势，本文算法由于采用模糊控制，小车行驶指令均由专家经验给出，因此会存在一些偏差，但偏差范围在可接受范围内；而直角转弯路径跟随的准确度则是本文的算法较好。

仿真实验中的计算耗时数据如表 4.4 所示。

表 4.4 仿真实验数据对比

	本文算法			基于最小二乘法线性拟合 的预瞄控制算法		
	采样次数	总耗时(ms)	平均耗时(ms)	采样次数	总耗时(ms)	平均耗时(ms)
方波	533	31257	58	504	35078	69
曲线	312	17788	57	313	19725	63

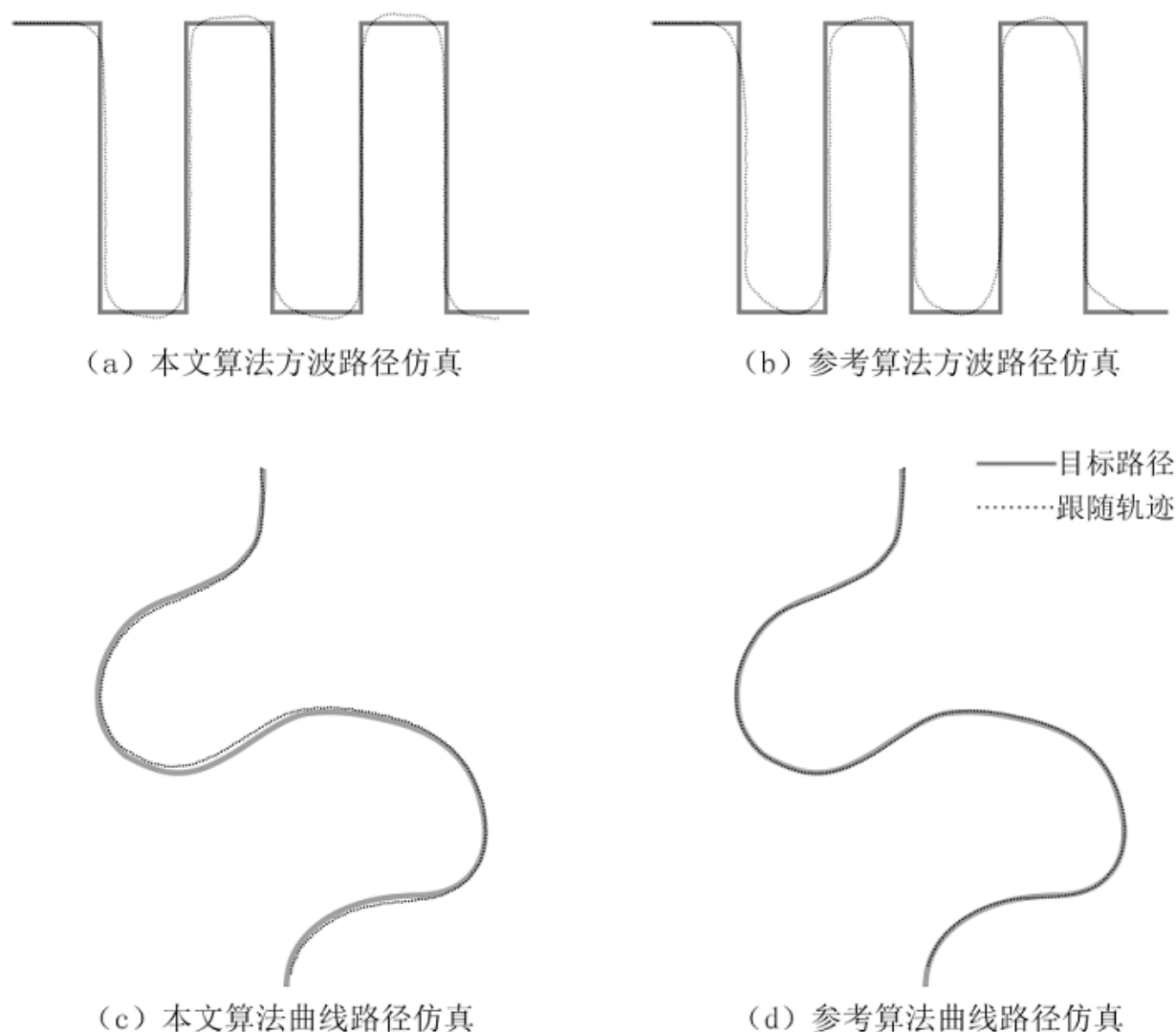


图 4.11 对比仿真实验结果

由仿真数据可知，本文提出的路径识别算法用时明显少于传统的最小二乘法的路径信息提取方法，约节约时间 10%。

当道路为曲率较小的曲线时，本文算法平均耗时 57ms，传统的最小二乘法路径信息提取方法平均耗时 63ms。当仿真路径为方波时，本文算法平均耗时 58ms，传统的最小二乘法平均耗时 69ms。原因是当采用最小二乘法拟合路径中心线时，直角转弯处的路径中心线为两条直线，因此计算量增加，相应的耗时增加，而本文的方法未受明显影响。在允许的路径跟随偏差范围内，本文提出的算法要快于基于路径边缘线提取和最小二乘法线性拟合的预瞄控制算法。

在仿真实验中，路径的图像信息是未经透视变形的平面图，提取道路中心线的路径识别算法需要在图像预处理的过程中将摄像机采集的透视图像逆变换得到鸟瞰图，而本文采用的方法直接通过方向码盘标定路径走向，进一步减少了计算量。由此可见本文路径识别的算法在计算量上要优于提取道路中心线的方法。

4.7 本章小结

本章详细介绍了基于图像识别和模糊控制的路径跟踪算法及其实现过程，通过方向码盘的设置，避免了路径信息的边缘提取，减少算法的计算量。并在 VS2010 环境下编写仿真程序，与传统的道路中心线提取以识别路径的预瞄控制算法做了对比，仿真结果表明，本文算法的计算量更小。

第 5 章 智能小车路径跟随试验

在行进过仿真实验后，本文自主搭建 20:1 的双轮驱动智能小车模型，独立完成嵌入式程序的开发设计，并在智能小车上验证算法的可行性。本章主要介绍智能小车的硬件平台搭建、嵌入式程序开发、智能小车硬件调试和实验结果及其数据分析。

5.1 硬件系统布置

在选好所有硬件后，还需要在小车上安装所有的元件，小车的结构参数经过测量如表 5.1。

表 5.1 小车各项结构参数

项目	参数
车轮轮距	150mm
摄像头离地高度	80mm
摄像头与地面角度	45°
摄像头图像起始线距小车运动中 心距离	20cm
最小转弯半径	75mm
NAER 码盘半径	50mm~75mm
过渡码盘半径	75mm~100mm
FAR 码盘半径	100mm~150mm

小车实物图如图 5.1 所示。



图 5.1 实验小车实物图

5.2 智能小车平台搭建

本节详细描述了系统的每个硬件模块，包括控制系统所选硬件的设计与引脚连接。稳定的硬件环境是软件能够良好运行的基础，所以硬件的选型与设计至关重要。

5.2.1 上位机配置

系统的上位机主要功能是开发控制台窗口程序，分析处理图像信息，发送和接收数据包。由于算法中涉及多线程计时延时串口数据发送，需要 CPU 可进行多线程运算。本文选用实验室的 PC 机，核心配置如下：

CPU: Intel(R) Core(TM) i5-2300 CPU 四核处理器，主频：2.80GHZ

内存：4GB

网卡：Realtek PCIe GBE Family Controller

操作系统：Windows7 64 位

上位机通过 USB 接口连接 USB 转串口线，转换线选用德飞莱 USB-TTL 模块，该模块使用 CH340 芯片，需要手动安装驱动，驱动软件为 USB-SERIAL CH340，安装成功后，设备管理器中会显示串口，本机串口号为 COM4，串口号因 PC 机和所连 USB 口不同会有所不同，可在设备管理器中自行查看。

5.2.2 移动端设计与搭建

移动端硬件系统主要由七部分组成，分别为智能小车底盘、电源模块、ARM 开发板、网络摄像头、串口通信模块、电机控制模块、车速检测模块，其中 ARM 开发板是智能车的核心控制平台，控制电机控制模块和车速检测模块，并将检测到的信息转发给上位机，以及接收上位机发送的速度控制信息，并转发给电机控制模块以驱动电机；网络摄像头是上位机搭载在移动端的眼，串口通信模块是上位机与智能车之间的通信桥梁，通过蓝牙通信协议与上位机数据互传。

5.2.2.1 车载控制板的选择

本文选用笔者现有的野火 STM32F103VET6 开发板作为智能小车开发板。

智能小车系统采用的 STM32F103VET6 属于 STM32 系列中的“增强型”系列，片内 FLASH 为 512KB，SDRAM 为 64KB，最高工作频率为 72MHz，数字芯片封装形式为 LQFP，针脚数为 100。芯片包含 3 个 12 位的 ADC、4 个通用 16 位定时器和 2 个 PWM 定时器，还包含 2 个 I²C 接口、3 个 SPI 接口、2 个 I²C 接口、1 个 SDIO 接口、5 个 USART 接口、一个 USB 接口和一个 CAN 接口。开发板集成了 STM32 芯片所需的外设电路，最多可输出 16 通道的 PWM 码，可使用 4 个通

用计时器、两个高级定时器和两个基本定时器，完全可以满足实验需求，因此，以 STM32F103VET6 作为本系统的车载核心处理器。开发板实物图如图 5.2 所示。

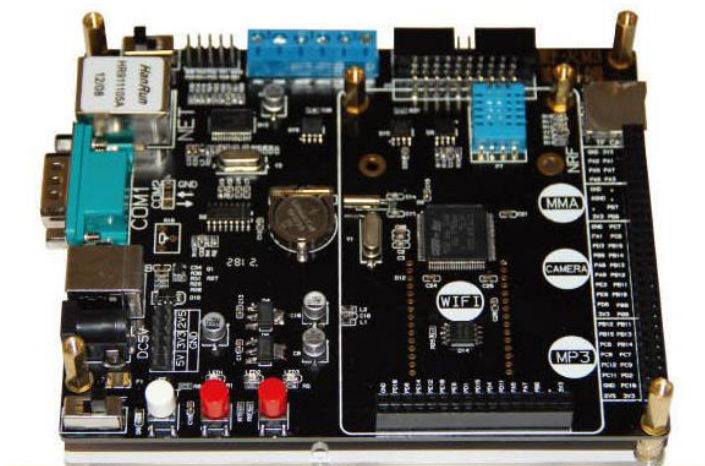


图 5.2 STM32 开发板实物图

5.2.2.2 蓝牙串口通信模块

本系统数据互传是由蓝牙串口通信模块实现的。蓝牙串口线是无线串口的一种，可代替现有的串口线。本系统选用串口模块为德飞莱 2.0 版蓝牙串口通信模块，基本参数如下：

串口模块工作电压：3.6~6V；

波特率：默认 9600，可选 4800,9600,19200,38400,57600,115200；

核心模块尺寸大小为 28mm x 15mm x 2.35mm；

工作电流：配对时 30~40mA，配对完毕未通信 2~8mA，通信中 8mA。

蓝牙串口通信模块实物图如图 5.3 所示。

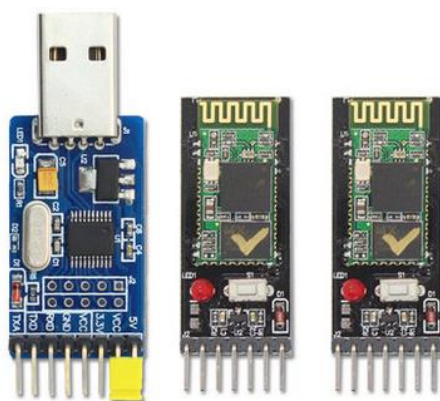


图 5.3 蓝牙串口通信模块实物图

图 5.3 中最左侧为 USB-TTL 模块，即 USB 转串口线。右侧两片为蓝牙串口模块，可分别配置为主机和从机。将主机与 USB-TTL 模块相连，连接方式如表 5.2 所示。

表 5.2 上位机端 USB-TTL 与蓝牙串口主机连接方式

PC (电脑) USB-TTL	蓝牙串口模块
VCC	VCC
GND	GND
RXD	TXD
TXD	RXD

本文中端口设置为 115200-N-8-1，即波特率为 115200，校验位无 (None)，数据位 8，停止位 1。配置好主从机后，将 USB-TTL 模块与电脑相连，从机连接 STM32 开发板，由开发板对其供电，供电电压为 5V。当主从机同时供电时，主机会自动搜索从机并自动配对，配对成功后，两个模块的 led 灯同步闪烁，先快闪 2 次，然后满闪 1 次，反复循环。此时，主机和从机均可发送和接收信号，接收到的信号由 TXD 口输出，从 RXD 口读取数据后发送。

STM32 的许多通用输入/输出接口 (GPIO) 是复用的，在配置 GPIO 口时，需要查看 STM32 用户手册，选择合适的 GPIO 口，在使用复用功能时，需要先把相应的 GPIO 口进行初始化。查询《STM32F10xxx 参考手册》(后称《手册》) 的引脚功能定义，如表 5.3 所示。

表 5.3 USART2 与 GPIO 口的连接与配置

复用功能	USART2_REMAP=0	USART2_REMAP=1
USART2_TX	PA2	PD5
USART2_RX	PA3	PD6

选定 PA2 作为串口 USART2 的数据发送引脚，PA3 作为数据接收引脚。

串口模块与 STM32 开发板的连接方式如图 5.4 所示。

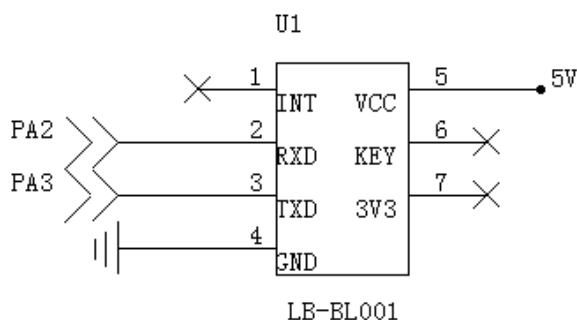


图 5.4 蓝牙串口模块线路连接图

在主程序 main() 中，调用函数 USART_Config() 函数来配置串口，该函数调用 USART2_Init() 和 NVIC_Configuration()。USART2_Init() 的部分代码如下：

```
RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO
```

```

RCC_APB2Periph_GPIOA, ENABLE);
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure); //初始化 PA2
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOA, &GPIO_InitStructure); //初始化 PA3
USART_InitStructure.USART_BaudRate = 115200; //设置波特率为 115200
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
//串口一次传输字长为 8 位
USART_InitStructure.USART_StopBits = USART_StopBits_1;
//停止位设置为 1 位
USART_InitStructure.USART_Parity = USART_Parity_No ; //无奇偶位校验
USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
//配置串口模式。为了配置双线全双工通讯，需要把 RX 和 TX 模式都打开
USART_Init(USART2, &USART_InitStructure); //向寄存器写入参数
USART_ITConfig(USART2, USART_IT_RXNE, ENABLE); //开启中断
USART_Cmd(USART2, ENABLE); //使能 USART2 外设
该函数作用为初始化 PA2 和 PA3 接口，配置通信协议为 115200-8-1-N。

```

NVIC_Configuration()的作用为初始化 USART2 的中断功能，配置中断优先级为 1，该优先级在本系统中为最高，可打断其他中断程序。

5.2.2.3 小车底盘

小车底盘选用双层亚克力板支架，层间用铜柱支撑。车轮直径 65mm，车轮驱动电机为直流减速电机，工作电压为 6~12V，转速随电压提高而提升，轮距 L 为 150mm，实物图如图 5.5 所示。



图 5.5 小车底盘实物图

5.2.2.4 电源模块设计

本系统选用由两节 18650 型号电池串联而成的 LIION 充电锂电池作电源，分别给开发板供电以及驱动电机。电池额定电压为 7.4V，在满电状态下为 8.2V 左右。

STM32 开发板默认供电方式为 USB 供电，USB 供电电压为 5V。该开发板供电电压不能高于 5V，否则会烧坏，因此，若要在使用 LIION 电池对 STM32 开发板供电，首先需要降压到开发板可以承受的电压 5V。本文采用 LM2940CT-5.0 低压差稳压器降压，该芯片工作温度为 $0^{\circ}\text{C}\sim 125^{\circ}\text{C}$ ，输入电压为 6.25V~26V，输出电压为 4.75V~5.25V，输出电流为 5mA~1A，能很好地满足 STM32 开发板的供电需求。线路连接图如图 5.6 所示。

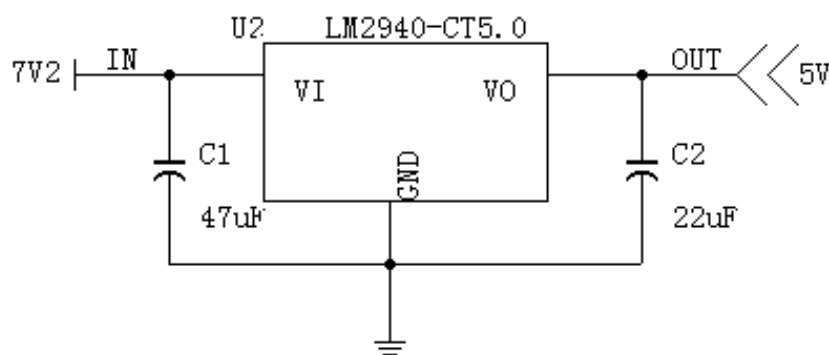


图 5.6 开发板电源模块线路连接图

在使用过程中，发现用单个 LM2940CT-5.0 稳压器为开发板供电时，发烫现象严重，故系统采用三个 LM2940CT-5.0 并联的方式，分散输出电流提高散热效率，同时起到保护稳压器的作用，避免造成不必要的损失。输出电压经实测为 5.0V。7.2V LIION 电池可对左右轮两路电机直接供电。

5.2.2.5 电机控制模块

小车左右轮分别由两只电机控制。电机控制模块是智能小车系统的执行机构，通过左右轮转速的调整和差速旋转，可使小车直行、转弯、原地绕圈等。本文采用 STM32 分别向左右轮电机输出两路单独的 PWM 码来控制电机。

开发板输出的 PWM 码的等效电压随着占空比的变化而变化，电压范围为 0~4.6V，而车轮电机的工作电压为 6~12V，故无法直接用 PWM 码驱动电机旋转，因此需要外接驱动电源，电源电压由前述知为 7.2V。开发板的 PWM 码输出端接三极管 B 极，构成开关电路，电机驱动电源串联开关电路后与电机相连，当 PWM 码处于高电平状态时，三极管开关导通，驱动电源电压直接作用于车轮电机上，驱动电机旋转；当 PWM 码处于低电平状态时，三极管开关断开，电机两端电压为 0。随着 PWM 在低电平与高电平之间不断切换，就在电机两端产生一个等效

电压，PWM 占空比越高，等效电压越高，占空比为 100%时，三极管开关常闭，电机两端电压为 7.2V，当占空比降低到一定程度时，电机两端等效电压不足以驱动电机旋转时，电机停转。

STM32 共有 8 个定时器，本文选择通用定时器 TIM3，利用 TIM3 产生两路 PWM 码来控制电机。查询《手册》后，选择 PA6 与 PA7 两个 GPIO 口，作为 PWM 码输出口。见表 5.4。

TIM3 功能产生 4 路独立的 PWM 码，本文选用通道 1 与通道 2 作为 PWM 生成通道。三极管开关电路如图 5.7 所示。

表 5.4 TIM3 与 GPIO 口的连接与配置

	TIM3_REMAP[1:0] = 00 (没有重映像)	TIM3_REMAP[1:0] = 10 (部分重映像)	TIM3_REMAP[1:0] = 11 (完全重映像)
复用功能			
TIM3_CH1	PA6	PB4	PC6
TIM3_CH2	PA7	PB5	PC7
TIM3_CH3	PB0	PB0	PC8
TIM3_CH4	PB1	PB1	PC9

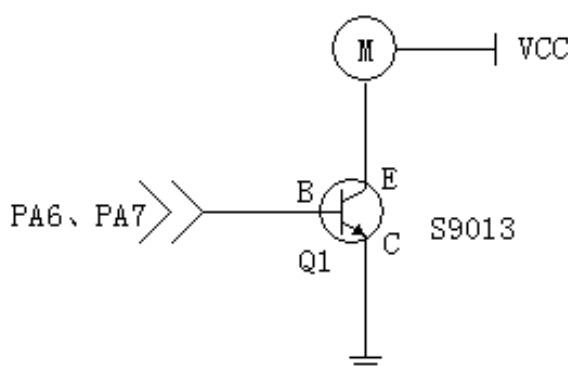


图 5.7 三极管开关原理图

B 极连接开发板的 PWM 信号发送 IO 口，接收到 PWM 信号时，当 PWM 信号在高电平时，三极管 EC 极导通，低电平时断开，驱动电机两端电压及等效电压如图 5.8 所示。

三极管选用 S9013 型 NPN 三极管，该三极管可承受集电极-发射极电压 25V，集电极-基极电压 45V，发射极-基极电压 0.7V，集电极电流 $I_{cMax}=0.5A$ ，工作温度 $-55^{\circ}C \sim +150^{\circ}C$ 。当电源充电结束时两端电压为 8.2V，试验用驱动电机在两端加载 8.2V 电压时，电流为 130mA，故三极管 S9013 可满足试验需求。

配置好串口后，当串口模块向开发板传入数据时，启动中断，中断程序调用函数 DRIVE_Getc (Res)，Res 为接收到的字符，DRIVE_Getc 函数根据接收到的字符，向电机输出控制信号。

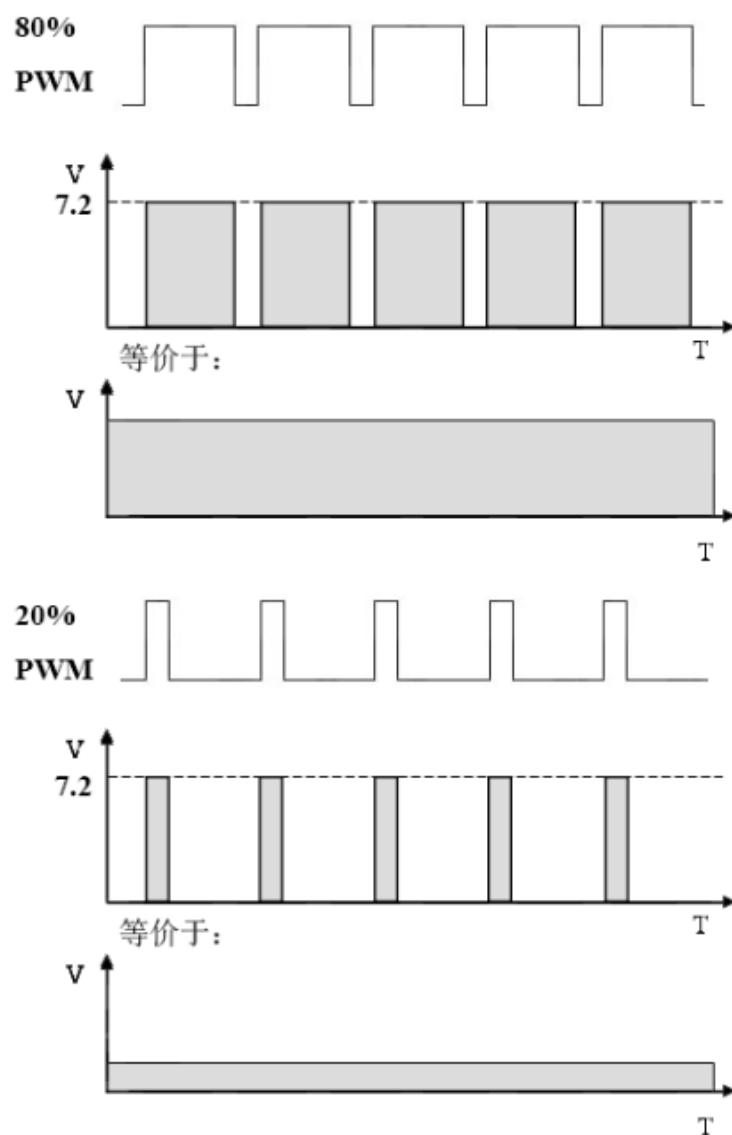


图 5.8 PWM 占空比及电机两端等效电压

部分代码如下：

```

TIM_TimeBaseStructure.TIM_Period = 999;
//当定时器从 0 计数到 999，即为 1000 次，为一个定时周期
TIM_TimeBaseStructure.TIM_Prescaler = 0;
//设置预分频：不预分频，即为 72MHz
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
//向上计数模式
TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = 0; //设置跳变值，当计数器计数到这个值时，
电平发生跳变，此处占空比设置为 0
TIM_OC1Init(TIM3, &TIM_OCInitStructure); //使能通道 1

```

```

TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Enable);
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = 0;
//设置通道 2 的电平跳变值，占空比设置为 0
TIM_OC2Init(TIM3, &TIM_OCInitStructure); //使能通道 2
TIM_OC2PreloadConfig(TIM3, TIM_OCPreload_Enable);
TIM_ARRPreloadConfig(TIM3, ENABLE); // 使能 TIM3 重载寄存器 ARR
TIM_Cmd(TIM3, ENABLE); //使能定时器 3

```

主程序调用 TIM3_PWM_Init()函数来初始化 PWM 模式和 GPIO 口，初始化完成后，PA6 和 PA7 输出占空比为 0 的 PWM 码，即处于待机状态。

5.2.2.6 速度检测模块

智能小车车轮转速检测模块的核心部件是 HC-020K 测速传感器。测速传感器是一款宽电压、高分辨率、短响应速度、开关量输出的测速模组，配合黑色码盘可以测量电机转速。码盘选用 20 线的码盘，车轮直径为 65mm，每当码盘转动一格，就可以计算得小车车轮驶过的距离为 $s=65*\pi/20=10.21\text{mm}$ 。该传感器为对射传感器，对射传感器和反射传感器相比，具有分辨率高、输出波形规整等优点。当激光被码盘遮挡时，传感器输出高电平，反之，输出低电平。传感器及 20 线码盘实物图如图 5.9 所示。

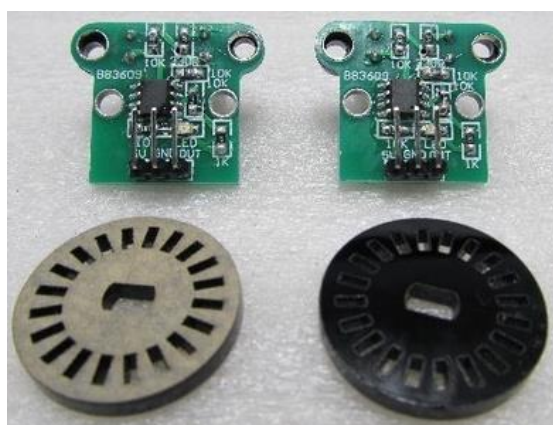


图 5.9 HC-020K 测速传感器与测速码盘模块实物图

该传感器工作电压为 4.5V-5.5V，测量频率为 100KHz，分辨精度为 0.01mm，由 LM2940CT-5.0 稳压管供电，输出端连接 STM32 开发板，配置 STM32 开发板与测速传感器连接的 I/O 口为上升沿捕获中断，即每当测速传感器输出端的电压由低升高是，触发中断。此时可用两种方式计算车速，一种是计数法，一种是计时法。计数法是在固定周期内（如 1s）统计触发中断的次数，根据中断次数计算车轮转动角度进而计算得到车轮转速，计时法则是计算相邻两次中断的时间间隔，此时车轮转过的角度固定，则可由此计算得到车轮转速。由于本文采用 PID 控制

算法来控制转速，对反馈速度的精度有较高的要求，因此采用计时法来计算车轮转速。两次中断的时间间隔若为 t ，即可计算出小车车轮当前转速，转速 $v=s/t$ 。

该传感器并未装配硬件防抖功能，因此需要在 STM32 开发板的车载程序中设计过滤程序，防止临界抖动。

本文选用通用计时器 TIM2 作为车载测速模块的信号接收计时器，查询《手册》，见表 5.5。

表 5.5 TIM2 与 GPIO 口的连接与配置

复用功能	TIM2_REMAP[1:0]=00 (没有重映像)	TIM2_REMAP[1:0]=01 (部分重映像)	TIM2_REMAP[1:0]=10 (部分重映像) ⁽¹⁾	TIM2_REMAP[1:0]=11 (完全重映像)
TIM2_CH1_ETR	PA0	PA15	PA0	PA15
TIM2_CH2	PA1	PB3	PA1	PB3
TIM2_CH3	PA2	PA2	PB10	PB10
TIM2_CH4	PA3	PA3	PB11	PB11

由于 PA2 与 PA3 这两个 GPIO 接口已被串口模块占用，从线路连接方面考虑，尽可能选择相邻的 GPIO 口，因此本文选用 PB10 与 PB11，于此对应需要初始化 TIM2 的第三通道和第四通道，并开启完全重映射。电路连接如图 5.10 所示。

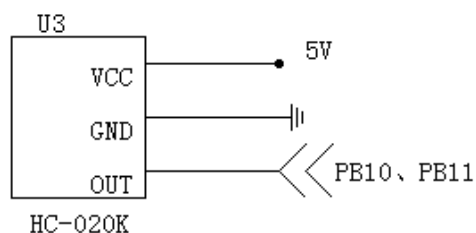


图 5.10 测速模块线路连接图

系统初始化时，配置的主工作频率为 72MHz。在对通用计时器 TIM2 的配置过程中，将预分频的值设为了 7200，由此可计算得 TIM2 的工作频率为 $72000000/7200=10\text{KHz}$ ，即每秒钟计数 10000 次，因此配置 TIM2 的计数周期为 10000，每当计数周期满 10000 产生溢出中断时，时间正好为 1 秒。

通用计时器 TIM2 的第三通道和第四通道采用相同的配置，分别检测左右轮转速，中断触发方式为上升沿触发，即当安转在电机上的码盘遮挡住 HC-020K 测速传感器的直射激光的瞬间触发计时器中断，且在计时器中断触发后的 15 个周期内，即 1.5ms 内不再触发中断，小车最高行驶速度 v_{max} 为 2m/s，已知车轮直径 d 为 65mm，测速码盘为 20 线，即车轮旋转一周，码盘可有效遮挡传感器激光 20 次，则两次遮挡的最短时间间隔 t_0 的计算方式为：

$$t_0 = \frac{d \times \pi}{20 \times v_{max}} \quad (5.1)$$

由式(5.1)得 $t_0=5.1\text{ms}$ ，因此 1.5ms 的过滤时间满足功能要求。且实验表明，该设置可有效地过滤测速传感器输出波形中的临界抖动，起到很好的防抖作用。

第三通道和第四通道的中断优先级均为 2，低于串口通信的中断优先级，因此，车轮的测速中断不会影响串口信号的接收。

主程序调用 TIM2_PB_Config()函数对 PGIO 口及中断触发方式、计时模式进行配置，函数部分代码如下：

```
TIM_TimeBaseStructure.TIM_Prescaler=7200-1;//预分频设为 7200
TIM_TimeBaseStructure.TIM_CounterMode=TIM_CounterMode_Up;
TIM_TimeBaseStructure.TIM_Period=10000-1;//计数周期设为 10000
TIM_TimeBaseInit(TIM2,&TIM_TimeBaseStructure);
TIM_ICInitStructure.TIM_Channel=TIM_Channel_3;
TIM_ICInitStructure.TIM_ICPolarity=TIM_ICPolarity_Rising;//上升沿触发
TIM_ICInitStructure.TIM_ICSelection=TIM_ICSelection_DirectTI;
TIM_ICInitStructure.TIM_ICFilter=0x0F;//延时 15 周期等待中断，软件防抖
TIM_ICInit(TIM2, &TIM_ICInitStructure);//打开通道 3
TIM_ICInitStructure.TIM_Channel=TIM_Channel_4;
TIM_ICInit(TIM2, &TIM_ICInitStructure);//打开通道 4
```

配置好中断触发模式后，还需要编写中断服务函数。PID 控制算法的实现在此中断函数中编写。

5.2.2.7 无线摄像头模块

智能小车系统选用手机摄像头作为局域网摄像头使用。在手机上安装 WebOfCam 程序，打开程序后，在上位机（PC）中打开浏览器，访问局域网 192.168.1.111:8080，即可在浏览器中实时显示手机传回的图像信息，延时大小根据手机型号的不同略有不同。本文选用 iphone6s，图像延时约为 50ms。

5.3 小车轮速调节

小车轮速控制是路径跟随实施的最后阶段，也决定了控制指令能否被准确地执行。由于本文的智能小车选用的是直流电机，而直流电机并不能根据电流大小精确地响应，因此需要对驱动电机的电压响应关系进行测量并通过调节使其能按照控制指令的要求稳定地输出转速。

5.3.1 驱动电机调试

小车驱动电机为直流电机，电机的转速与扭矩与电机两端加载的电压高度正

相关。当电压从零逐渐升高时，电机起转电压为 V_s ，当电机两端电压由高降低，电机停转电压为 V_e ，且有：

$$V_s > V_e \quad (5.2)$$

将电机按图 5.7 连线方式连接开发板，电源电压为 7.2V，电机上安装小车车轮和测速码盘，在小车整装的情况下检测车轮转速，并将转速换算成车轮旋转的线速度，在输出 PWM 码占空比 70%-100% 范围内，对电机转速采样并作多项式线性拟合，如图 5.11 所示。

STM32 开发板对三极管基极输出占空比为 70%~100% 的 PWM 码，图 5.11 横轴为 PWM 码占空比(%)，纵轴为根据检测数据计算得到的小车车轮旋转线速度，单位为 m/s，图中所示 P2 点为小车整装时车轮起转点，P1 为小车整装时车轮减速时的停转点。由图可知：当系统设定的车轮线速度小于 0.6m/s 时，若此时小车

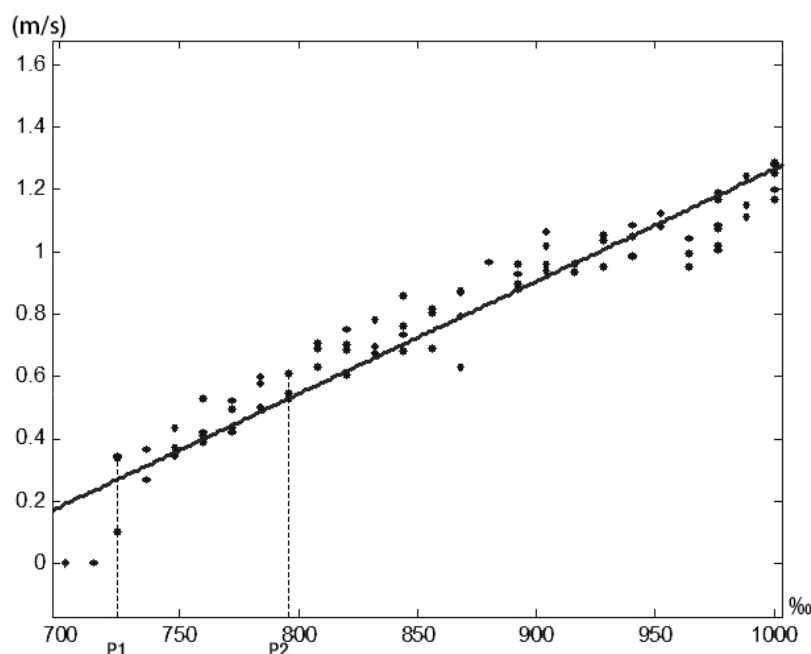


图 5.11 PWM 码与车轮空转线速度

处于待机状态，则电机无法起转，因此需要向电机传送一个瞬间的高占空比 PWM 码来启动电机；电机转速与 PWM 占空比成正相关，但并不完全呈线性关系，且不同直流电机间也有略微差别，因此若要较精确地控制电机转速，就需要采用闭环系统来控制，比如本文所采用的 PID 闭环控制。

由于本文采用 PID 控制，因此不需要对电机参数有详细的测量与计算，由实验测量可知，小车整装时车轮旋转线速度的可控范围为 0.3m/s~1.2m/s。小车待机状态下的启动时，若向电机控制模块输入的启动速度为 0.4m/s，则相应的占空比不足以使电机开始转动，因此在电机停机状态启动时，开发板瞬间输出占空比为 85% 的 PWM 码，之后 PWM 码占空比的设定则交给 PID 控制程序。

5.3.2 轮速 PID 调节

PID 控制算法是最古老并广泛用于工业生产过程中的控制算法，其控制过程由三部分组成，分别为比例（P）、积分（I）、微分（D）控制。当用 PID 控制算法驱动直流电机是，其输出的比例部分主要反映了实际速度与目标速度的误差，是成正比的；输出的积分部分则与误差的积分成正比，控制速度变化程度；输出的微分部分是误差的变化速率，反映了速度的变化趋势。PID 控制算法不需要完全掌握被控对象的结构和参数，也不需要得到精确的数学模型，使用起来十分方便^[36]。

PID 控制并非一个整体，实际上 P、I、D 是可以独立使用的，当然也有 PI 和 PD 控制，可以依据不同的现实需求灵活选择相应的控制算法，PID 控制器组成如图 5.12 所示。

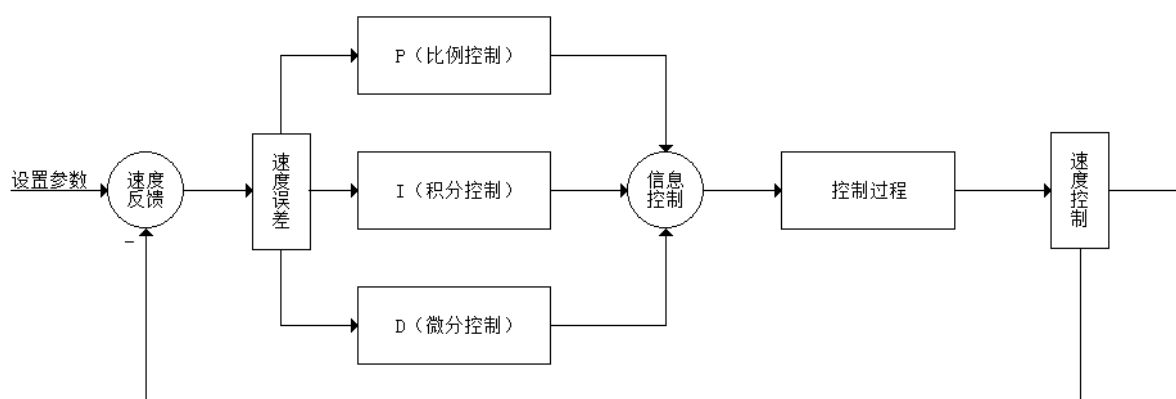


图 5.12 PID 算法控制器组成图

1) 比例控制部分：对于很多控制应用而言，在 0 与固定的控制信号之间的突变不能产生平滑的控制效果，可以通过使用线性或是比例项予以代替来加以改进，而且目标速度与实际速度的差值作为误差，改变比例项系数将改变控制器效果，取值越高，控制器相应越快；然而取值过高会使系统产生震荡，因此需要选择合适的比例系数，保证系统快速相应的同时，也要避免系统超调过大，以至于产生震荡。

2) 积分控制部分：积分控制部分很少单独使用，大部分都是与 P 控制器或 PD 控制器配合使用。积分控制器相当于延后控制，用以消除 P 控制器产生的稳态误差。

3) 微分控制器：微分控制器也很少单独使用，主要与 P 控制器和 PI 控制器配合使用。微分控制器相当于超前控制，用以改善 P 控制器对输入变化的响应，减少 P 控制器带来的超调，使系统更快地到达稳态^[37]。

本文在控制小车驱动轮电机时，使用 PD 控制，控制器的控制信息为开发板

输出的 PWM 码的占空比，反馈值为小车车轮的转动线速度。中断服务函数中 PID 控制部分流程如图 5.13 所示。

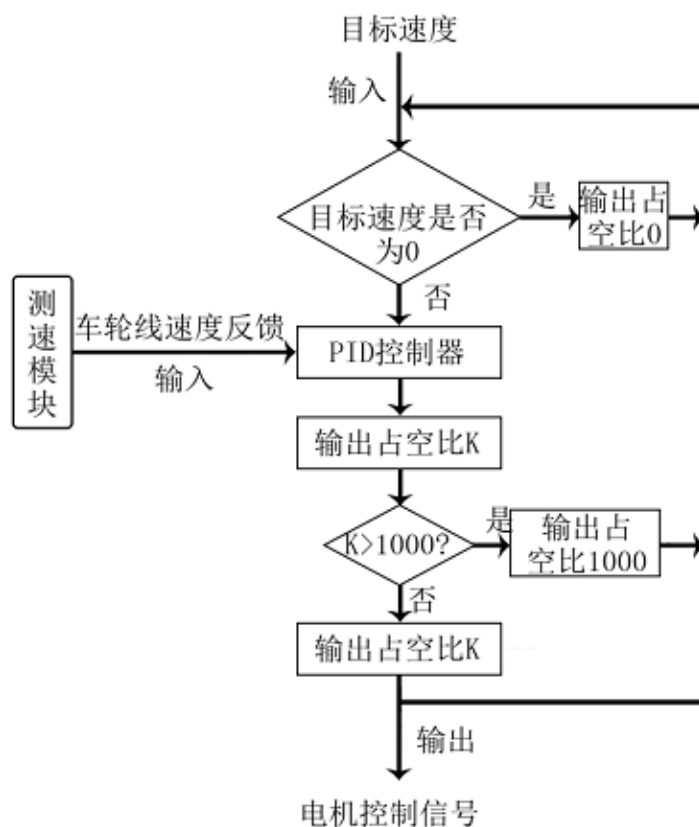


图 5.13 PID 控制算法流程图

PID 控制过程中，PWM 码占空比的表示方式为 $K\%$ ，当 $K>1000$ 时，电机转速已不再增加，故最高输出 K 值为 1000。PID 控制器计算占空比 K 的方法如下：

$$K = \left(t - \frac{\pi d}{20 \cdot v_g}\right) \cdot kp + (t - 2t_{p1} + t_{p2}) \cdot kd + K_p \quad (5.3)$$

其中： kp 、 kd 分别为比例控制系数和微分控制系数；

v_g 为车轮目标线速度；

d 为车轮直径；

t 为当前检测周期检测到的两次中断的间隔时间；

t_{p1} 、 t_{p2} 分别为为上一检测周期和上上检测周期的两次中断间隔时间；

K_p 为当前的 PWM 码占空比。

5.4 指令延时及方向修正

完成图像的分析计算后，剩下的就是将指令发送给移动端控制板。本文通过无线蓝牙串口实现上位机与移动端的数据收发，而串口发送指令的时间和移动端执行指令的时间也需要加以控制。

小车在行驶过程中,摄像头拍摄到的图像起始线是小车前方一定距离的地面,设车轮与图像起始线的地面距离为 S ,小车移动速度为 v ,则在摄像头形成图像并开始回传到小车车轮行驶到需要执行对应指令之间就有时间间隔 T ,根据第四章小车执行指令时刻的讨论,时间间隔可以近似为

$$T = \frac{S - R_M}{v} \quad (5.4)$$

其中 R_M 为方向码盘的半径。

为方便软件实现,本文设定移动端收到行驶指令后立即执行,上位机在计算得到对应指令时,等待时间 t_0 后再向移动端发送行驶指令, t_0 的计算方式如下:

$$t_0 = \frac{S - R_M}{v} - t_1 - t_2 - t_3 \quad (5.5)$$

其中, t_1 为主函数中截取图像并对图像作分析计算所消耗的时间, t_2 为网络摄像头从拍摄到图像信息在显示器中显示的时间, t_3 为上位机向串口写入指令到移动端控制板接收到指令并执行所消耗的时间。 t_1 的大小,每个计算周期会有略微区别, t_2+t_3 的大小由实验确定。

根据以上分析,上位机在行驶指令的生成到发送需要一个短时间的延时,但如果在主函数中等待,会直接增加一个指令周期的时长,对小车路径跟随的准确性有很大影响,甚至直接导致路径跟随失败,因此,本文采用设立指令缓存区并开多线程计时的方法来确保行驶指令的准确发送。上位机在图像分析处理完成后,将行驶指令和计算耗时一并存入缓存区中,副线程读取缓存区中的指令后,根据公式(5.5)计算等待时间,计时结束则发送指令,指令的发送亦在副线程中实现。

小车指令延时发送和方向修正如图5.14所示。

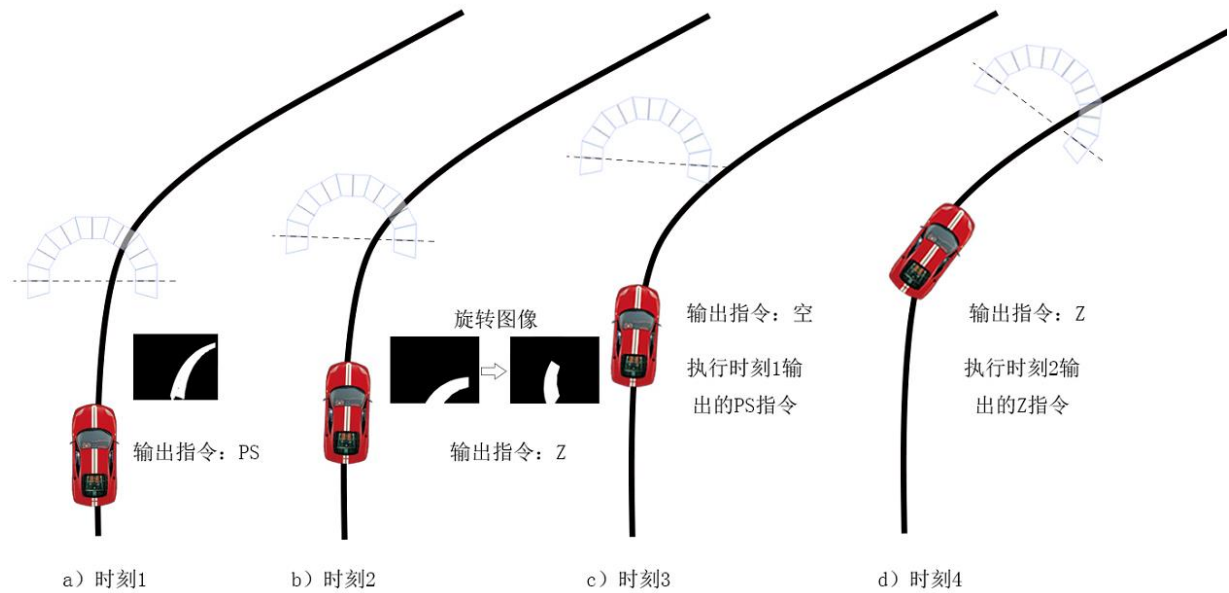


图 5.14 指令延时发送和方向修正示意图

在时刻1，小车获取图像信息提取路径特征，输出指令PS，指令被存入指令缓存区。在时刻2，小车根据提取的路径信息可得到PM指令，但由于时刻1的指令尚未被执行（执行后小车的朝向将与路径方向趋同），若输出PM指令，小车会在执行时刻1的指令后继续执行PM指令，小车会偏离目标路径，因此，需要对时刻2获取的图像信息作方向修正，如图5.14所示时刻2输出的行驶指令为Z，即小车向正前方行驶。到时刻3，上位机负责发送指令的副线程结束等待时间，向小车发送时刻1生成的PS指令，与此同时，开始采集图像信息。然而时刻3采集的图像中却无路径信息，于是生成一个空指令存入缓存区，此时停机计数器开始计数，当计数器的值达到3时即向小车发送停机指令。时刻3产生的空指令的作用复制上一指令，在当前例子中就是时刻2生成的Z指令。到时刻4时，小车开始执行时刻2生成的Z指令，且在图像处理时检测到路径信息，于是将停机计数器清零，等待下一次计数的开始，并在路径特征提取后输出Z指令。

方向修正角度与上一输出指令的对应关系由实验得出，且速度不同时方向修正角度也不同，记逆时针旋转为正，修正角度见表 5.6。

表 5.6 修正角度与上一行驶指令关系

行驶速度 (m/s)	NB	NM	NS	Z	PS	PM	PB	(deg)
0.4	—	58	-22	0	22	58	—	
0.6	—	—	-42	0	42	—	—	

5.5 实验

本文所采用的方法在图像处理过程中采用的滤波方式为腐蚀、膨胀，以及对路径信息的兴趣区域提取，用以消除没有与路径相连大面积深色区域，固在实验环境选取时需要选择浅色背景色的平坦地面，且没有大面积深色区域与路径交叉。实验道路如图 5.15 (a) 所示。将路径作逆透视变化得到俯视图，并通过图像处理消除背景噪音，得到 5.15 (b)。

设定小车低速状态时运动中心线速度为 0.4m/s，高速状态运动中心线速度设置为 0.6m/s，小车各项参数由表 5.1 给出，小车由 A 点向 B 点行驶。实验过程中，车载控制板通过无线蓝牙串口向上位机发送计算后的小车左右轮线速度，速度采样周期为 0.15s。上位机从蓝牙串口接收到数据后，将数据存储在文本文件中，留作分析，速度数据见附录 B。实验中小车顺利从 A 点到达 B 点。

实验结束后，对上位机接收到的左右轮线速度数据做矢量积分，并在与图 5.15 (b) 尺寸相同的空图中标出每一速度采集时刻的小车运动中心位置，并与图 5.15 (b) 做对比，如图 5.16 所示。

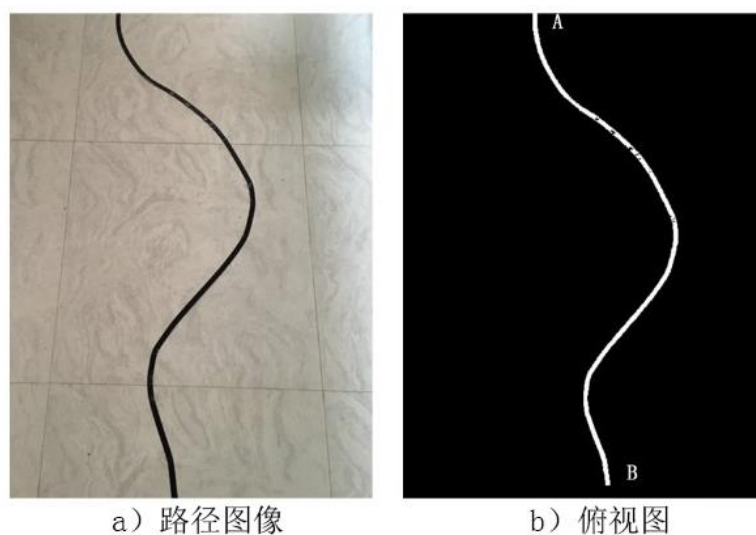


图 5.15 实验路径

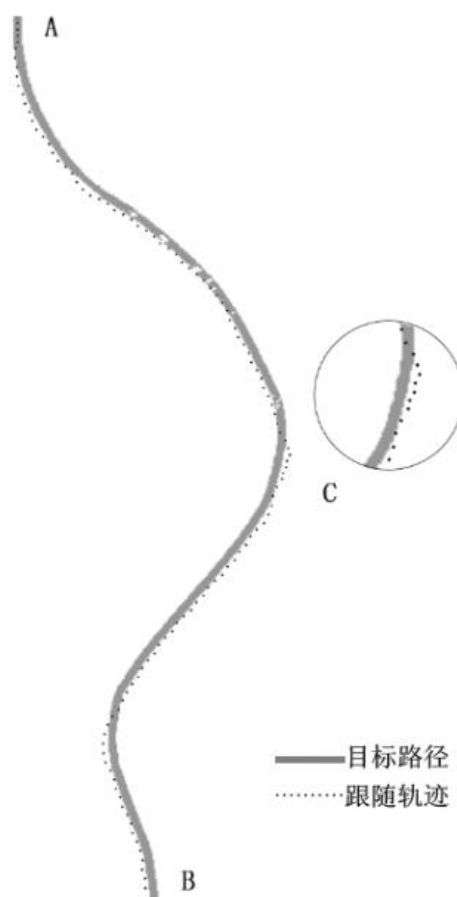


图 5.16 目标路径俯视图与小车运动中心位置对比图

结果表明，本文搭建的智能小车可接受上位机的控制，在误差允许范围内，沿目标路径行驶。启动点 A 点附近小车位置较密，表明小车启动时以低速启动，在检测到前方路径偏角较小时，进入高速状态；C 点附近小车运动中心位置较密，表明小车在此处速度降低，这说明小车在行驶过程中可以实现高速状态与低速状态的切换。

总结与展望

本文以智能小车为研究对象，主要研究智能车的环境感知和路径跟随问题。本文以计算机视觉作为智能小车识别路径的方式，通过研究路径特征提取的算法，在已有算法的基础上改进创新，减少路径特征提取的计算量，并与其它算法作对比来验证本文算法的高效性。路径跟随控制采用模糊控制，并在自主搭建的智能小车模型上验证控制算法的有效性。

1 主要的研究工作和创新点

1) 根据智能车路径识别及路径跟随的需求，设计智能车系统并搭建硬件结构。已 STM32F103 开发板为开发平台，构建嵌入式硬件控制系统，搭建了 20:1 的双轮差动小车模型，用以验证算法。

2) 建立车辆运动学模型，为后续的算法控制及实验仿真和结果分析打下基础。研究了智能小车路径跟随的算法，提出了一种基于图像识别的远程闭环路径跟随控制算法，并用模糊控制算法实现了智能车的路径跟踪。采用方向码盘比对的方法减少了路径识别过程中的计算量

3) 在 VS2010 环境中编写仿真程序及实验小车控制程序。在小车的控制过程中，与以往的单周期完成图像采集处理、路径识别、指令生成及发送不同，本文采用多线程控制程序，在完成一次图像采集处理、路径识别、指令生成的周期后，就将指令存入缓存区，继续下一周期的图像识别过程，而存入缓存区的指令由副线程延时发送。这样在整个小车操纵过程中，可以控制行驶指令的执行时间，又不影响下一周期的图像识别，使整个控制过程有更强的连续性。

2 工作展望

通过图像处理的方式识别车道线，而不是求解车道中心线代数解并计算曲率来获取行驶指令，更符合人类的识别习惯。随着计算机技术的飞速发展，计算机的运算速率也在快速提升，“性能过剩”使得通过图像识别来路径跟随开始成为可能。

在本文工作的基础上，还可以进一步研究如下问题：

1) 考虑多摄像头信息采集，增加移动端小车的视野范围，以便更全面、更顺

畅地控制小车精确地沿着给定路径行驶。

2) 更换本文采用的直流驱动电机, 替换以步进电机来驱动小车前行。步进电机相对直流电机有较大的优势, 首先步进电机的控制更为精准, 不会受到驱动电源电压的波动影响, 虽然用直流电机驱动时采用了 PID 控制, 但 PID 控制仍然需要一些调整时间, 这会引起小车的误差累计。

3) 本文的图像获取方式依然有待优化, 包括传输图像的格式以及传输的途径。传输图像的格式直接影响到传输数据量的大小, 而传输途径则决定了传输的速率, 由于本文的算法不需要高分辨率的图像信息, 甚至可以先将图像处理为占用空间较小的格式和较低的足以识别道路的分辨率再进行图像传输, 这不仅减少传输图像的耗时, 还对减少后期上位机的计算量有很大的作用。

4) 本文图像处理的方式和过程依然有待优化, 图像处理耗时还有较大的缩减空间, 这可为多摄像头图像信息采集预留更大的空间, 也可以更好地保障基于图像识别的路径跟随控制算法的实时性。

参考文献

- [1]. Willie D Jones. Keeping cars from crashing IEEE Spectrum,2001,38(9):40-45
- [2]. 牛学军, 朱茵, 楼涛.道路交通安全风险预管理对策.综合运输, 2008, (10): 42-46
- [3]. 中华人民共和国国家统计局.中国统计年鉴交通事故情况(2014).
<http://www.stats.gov.cn/>. 2015-10-17 徐利民.智能交通系统发展及其战略思考.
技术经济, 2003,4:15-16
- [4]. Kyongsu Yi,Minsu Woo,Sung Ha Kim,et. al. An Experimental Invertigation of a
CW/CA System for Automobiles Using Hardware-in-the-Loop Simulations. In
Proceeding of the American Control Confrennce. New Jersey,USA, 1991,
1:724-728
- [5]. Intelligent Vehicle-Highway Society of America. Strategic Plan for Intelligent
Vehicle-Highway Systems in the United States. Washington, D.C. , 1992, 34-56.
- [6]. 孙振平, 安向京, 贺汉根. CITAVT-IV 视觉导航的自主车.机器人, 2002,
24(2):115-121.
- [7]. 党宏社, 韩崇昭, 段战胜.智能车辆系统发展及其关键技术概述.公路交通科
技, 2002, (4): 127-128
- [8]. 孔凡忠, 李克强.智能汽车的感念、架构、发展现状及趋势.中国汽车报,2014,7:
18-24.
- [9]. 李斌, 王春燕, 吴涛.中国智能公路磁诱导技术研究进展.公路交通科技, 2004,
21(11): 66- 69.
- [10]. 杨殿阁, 李克强.智能交通系统中的汽车技术.汽车工程, 2003, 25(3): 223- 226.
- [11]. Tsugawa S.Vision-Based Vehicles in Japan: Machine Vision Systems and
Driving Control Systems. IEEE Trans. on Industrial Electronics.USA.1994: 398-
405.
- [12]. Andresen E P. Visual algorithms for autonomous navigation. Proc. of IEEE
International Conf. on Robotics and Automation. USA. 1985: 856-861.
- [13]. Bertozzi M, Broggi A, Fascioli A. Vision-based intelligent vehicles: State of the
art and perspectives, Robotics and Autonomous Systems, 2000, 32 (1): 1- 16.
- [14]. Baten S.Techniques for autonomous off-road navigation, IEEE Intelligent
systems and Their Applications, 1998, 13(6): 57-65.

- [15]. Brusaglino G. Safe and effective mobility in Europe -the contribution of the PROMETHEUS programme. IEEE Colloquium on Prometheus and Drive . USA. 1992: 101-110.
- [16]. Broggi A. Automatic Vehicle Guidance. USA: World Scientific Publishing Company, 1999.
- [17]. Massimo Bertozzi, Alberto Broggi. GOLD: A Parallel Real-Time Stereo Vision System for Generic Obstacle and Lane Detection. IEEE Trans. On Image Processing, 1998, 7(1): 62- 82.
- [18]. Shladover S E. Automated vehicle control developments in the PATH program . IEEE Trans. on Vehicular Technology, 1991, 40(1): 114- 130.
- [19]. Hiteheok A. Configuration and Maneuvers in Safety-consciously Designed AHS Configuration . USA: University of California (Berkeley), 1995.
- [20]. RAN B, L H X. Development of vision-based vehicle detection and recognition system for Intelligent vehicles. 1999 TRB Annual Meeting. University of Wisconsin at Madison: Madison, 1999: 12-25.
- [21]. Ross A, Knepper, Matthew T M. Path Diversity Is Only Part of the Problem. IEEE International Conf. on Robotics and Automation. Japan, 2009: 3224- 3230.
- [22]. Shoemaker C M, Bomstein J A. Overview of the Demo III UGV Program. Part of the SPIE Conf. on Robotic and Semi-Robotic Ground vehicle Technology. 1998:202- 211.
- [23]. Gage D W. Current Thrusts in Ground Robotics: Programs, Systems, Technologies, Issues. Fourth Int'l Sym. On Technology and the Mine Problem, 2000. 314-350.
- [24]. Joint Robotics Program Master Plan, FY 2005, Office of the Secretary of Defense, USA, URL: <http://www.jointrobotics.com/>. 2005.
- [25]. Thrun S. Stanley: The Robot That Won The DARPA Grand Challenge. Journal of Field Robotics, 2006:112-153.
- [26]. 张瑞成.基于图像识别的多智能车跟随与防撞控制系统研究.[重庆大学硕士学位论文].重庆: 重庆大学.2011:7-45.
- [27]. Li Q, Zheng N, Cheng H. Springrobot: a prototype autonomous vehicle and its algorithms for lane detection. IEEE Trans. on Intelligent Transportation Systems, 2002, 5(4): PP.300-138.
- [28]. 王家恩, 陈无畏, 王檀彬等. 基于期望横摆角速度的视觉导航智能车辆横向控制. 机械工程学报, 2012, 48(04): 108-115.
- [29]. 王荣本.一种视觉导航的实用型 AGV 设计.机械工程学报, 2002, 38(11):

- 135-138.
- [30]. 刘蕊.智能车路径跟踪及其底层控制方法研究.[北京工业大学].北京: 北京工业大学.2013:2-10.
- [31]. 王慧.智能车的路径识别与控制策略的研究.[吉林大学硕士学位论文].吉林: 吉林大学.2009:10-15.
- [32]. 李进. 视觉导航智能车辆的路径识别和跟踪控制:[合肥工业大学博士学位论文].合肥: 合肥工业大学.2008:25-64.
- [33]. 汪明磊. 智能车辆自主导航中避障路径规划与跟踪控制研究.[合肥工业大学博士学位论文]. 合肥: 合肥工业大学.2013:52-64.
- [34]. 王志飞.基于 PID 驱动控制算法的智能车远程闭环控制的研究与实现.[吉林大学硕士学位论文].吉林: 吉林大学.2015:3-5
- [35]. 孙振平, 安向京, 贺汉根.CTIAVT-IV-视觉导航的自主车.机器人,2002,24(2):115-121.
- [36]. 孙敏敏.数据与模型融合的无模型改进控制算法研究.[中北大学硕士学位论文].广州: 中山大学,2014: 45-53.
- [37]. 唐振浩,唐立新,杨阳.基于数据驱动和智能优化的高炉十字测温温度预报.信息与控制, 2014, (3): 23:-36
- [38]. 谭晓波.摄像机标定及相关技术研究.[国防科技大学工程硕士学位论文].长沙: 国防科学技术大学研究生院.2004:13-20.
- [39]. G.Q. Wei and S.D. Ma. Implicit Camera Calibration: Theory and Experiment. IEEE Trans. on Pattern Recognition and Machine Intelligence, 1994.16(5):469-480.
- [40]. C.Delherm, J.M. Lavest and M.Dphome et al. Dense reconstruction by zooming. In B. Buxton and R. Cipolla, editors, Proc. Of the 4th European Conference on Computer Vision(ECCV'96), Springer-Verlag, Cambridge, Englannd, 1996.2:427-438.
- [41]. 朱大奇,颜明重.移动机器人路径规划技术综述.控制与决策,2010,(7):961-967.
- [42]. 李宝霞.基于模糊算法的移动机器人路径规划.[燕山大学工学硕士论文].秦皇岛: 燕山大学信息科学与工程学院.2010: 20-56.
- [43]. Lousia Lam, Seong W L, Ching Y S. Thinning methodologies-A comprehensive survey. IEEE Trans. Patt Anal. Machine Intell, 1992(14);869-885
- [44]. 沈俊, 葛燧和.计算机系统上的 YUV.计算机工程.2000,26 (1): 95
- [45]. 上海师范大学数学系编.回归分析及其试验设计.上海: 上海教育出版社, 1978.
- [46]. Kehtarnavaz N, Grisworld N, Lee J. Visual Control for an Autonomous Vehicle (BART)-the Vehicle Following Problem. IEEE Transactions on Vehicular

Technology:1991,40(3):654-662.

- [47]. Fujimori A, Nikiforuk P N, Gupta M M. Adaptive Navigation of Mobile Robots with Obstacle Avoidance. IEEE Transactions on Robotics and Automation. 1997,13(4):596-601

致 谢

研究生的求学生涯即将画上句号。回首这一段路程，虽然有些曲折，但是在很多人的帮助之下顺利的走完了，在论文完成之际，在此表示深深的感谢。

首先，感谢我的导师宋晓琳教授。一直以来，宋老师严谨治学态度和精益求精的工作风格都对我有着深深的影响，让我在做研究的过程中时刻提醒着自己要有高目标、高要求。在学习生活中，宋老师都是平易近人的。在遇到学习问题时，宋老师会指出问题的关键，引导我去解决问题。在不断的发现问题和解决问题的过程中，教会我做研究的关键：多看文献、多交流，拓宽思路。宋老师对我学业上的指导和帮助、与我推心置腹的交谈，我必当铭记在心！师恩难忘，在此，向宋老师致以崇高的敬意和衷心的感谢！。

此外，感谢实验室所有的师兄弟们。与你们相处的时间不多，但都轻松愉快。特别要感谢曹昊天师兄对我的关系和帮助，相识已有五载，未来路会更长，衷心祝愿曹昊天师兄在今后的求学和生活中一帆风顺，精彩不断。

同时，感谢母校的培养，感谢机械与运载工程学院的领导和老师们的关怀，千年学府，百年机械，感谢有你。

最后，感谢父母家人对我学业的支持和理解，并一直在我身后默默地付出，你们是最坚强的后盾。特别要感谢我的女朋友郑琪瑶在我研究生期间给予我的关心和包容。有了你们的支持，我才能走得更远。此恩无以回报，今后我将更加努力！

附录 A

作者在攻读学位期间发表的论文目录

- [1]. 宋晓琳，周南，黄正瑜等.改进 RRT 在汽车避障局部路径规划中的应用.湖南大学学报（自然科学版）.2016（已录用，收稿编号 jg20160115004）.

附录 B 实验数据

小车低速设为 0.4m/s，高速为 0.6m/s，采样周期 T 为 0.15s，共采集 113 组数据如下，每组数据中前一数据为左轮线速度，后一数据为右轮线速度，单位均为 m/s。

[illegible]

小车低速设为 0.4m/s，高速为 0.8m/s，采样周期 T 为 0.15s，共采集 113 组数据如下，每组数据中前一数据为左轮线速度，后一数据为右轮线速度，单位均为 m/s。

$$\{0.4, 0.4\}, \{0.4, 0.4\}, \{0.41, 0.39\}, \{0.42, 0.38\}, \{0.37, 0.43\}, \{0.77, 0.83\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.77, 0.83\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.77, 0.83\}, \{0.82, 0.78\}, \{0.82, 0.78\}, \{0.77, 0.83\}$$

, {0.82,0.78}, {0.82,0.78}, {0.77,0.83}, {0.82,0.78}, {0.82,0.78}, {0.82,0.78}, {0.82,0.78}, {0.82,0.78}, {0.81,0.79}, {0.81,0.79}, {0.8,0.8}, {0.8,0.8}, {0.78,0.82}, {0.83,0.77}, {0.83,0.77}, {0.83,0.77}, {0.83,0.77}, {0.83,0.77}, {0.78,0.82}, {0.78,0.82}, {0.78,0.82}, {0.83,0.77}, {0.78,0.82}, {0.83,0.77}, {0.78,0.82}, {0.78,0.82}, {0.83,0.77}, {0.78,0.82}, {0.78,0.82}, {0.83,0.77}, {0.83,0.77}, {0.8,0.8}, {0.77,0.83}, {0.77,0.83}, {0.77,0.83}, {0.64,0.96}, {0.37,0.43}, {0.42,0.38}, {0.43,0.37}, {0.38,0.42}, {0.43,0.37}, {0.83,0.77}, {0.83,0.77}, {0.78,0.82}, {0.78,0.82}, {0.78,0.82}, {0.83,0.77}, {0.83,0.77}, {0.78,0.82}, {0.83,0.77}, {0.83,0.77}, {0.78,0.82},