基于机器视觉的车载安全机器人的设计与实现

**摘 要**

随着人类汽车工业的蓬勃发展，汽车已经成为大众交通工具，数量巨大。汽车工业的发展便利了人们的生活，但有时也会对大众的生命财产安全带来威胁。车辆在高速行驶遇紧急情况停车时，需在车辆的后方规定距离的位置放置警示牌，但放置警示牌的过程有时是极度危险的。本文通过设计实现基于机器视觉的可自动巡航并测距的车载安全机器人，旨在为司机保驾护航，最大限度保障安全，降低由于高速追尾造成的交通事故的发生率。本文的主要研究工作为如下四个方面：

1. 设计机械结构，并使用步进电机编程实现车载机器人拍摄杆的折叠收纳。

(2)将基于Intel处理器的原车载安全机器人移植到树莓派上，调试实现机器人自动巡航到指定位置并停止。

(3)设计基于机器视觉的测速测距算法使得机器人可以对前方来车进行测距测速，用以判断机器人和车辆的安全状态。

(4)研究实现车载安全机器人与司机安卓手机在150米范围内的无线通信。研究并实现2.4GHz通信协议，编写安卓app完成机器人发出数据的接收。

**关键词** OpenCV 树莓派 安卓 2.4Ghz通信

**DESIG N AND IMPLEMENTATION OF**

**VEHICLE SAFETY ROBOT BASED ON MACHINE VISION**

**ABSTRACT**

As the fast development of the human automotive industry, cars have become one of the most important public vehicle with a huge quantity. It’s obvious that the automotive industry has brought us great convenience, whilst the potential safety hazard is brought simultaneously. The highway is the place where accidents happen frequently. The driver has to put a warning board 150 meters behind the car when emergency happens leading to emergency shut-down. However the board putting procedure is extremely dangerous. In this paper, we are going to describe the design and implementation of vehicle safety robot based on machine vision, aiming to keep the safety for the drive and the passengers and reducing the incidence rate maximally. The main work of this paper is as follows:  
 (1) Design the Mechanical structure and implement the folding and storage of the photo rod by programming and control the stepper motor.

(2) Transplant the source project base on Intel core to the Raspberry Pi, debug it until the car can be cruise controlled.

(3) Design and implement distance and velocity measurement algorithm for the car to measure the velocity information of the coming cars, use which the robot can evaluate the security state in real time.

(4) Research and implement the wireless communication between the robot and the Android phone of the driver within a distance of 150 meters. Study the 2.4GHz communication protocol and write an Android application the receive the information sent by the remote robot.

**KEY WORDS** OpenCV Raspberry Pi Android 2.4GHz communication

**目　　录**

[第一章 绪论 1](#_Toc516516636)

[1.1 研究背景 1](#_Toc516516637)

[1.1.1 我国当今道路交通安全现状 1](#_Toc516516638)

[1.1.2 我国高速公路相关道路交通法规 1](#_Toc516516639)

[1.1.3 图像识别技术与自动驾驶 2](#_Toc516516640)

[1.1.4 安卓App开发现状 2](#_Toc516516641)

[1.2 研究内容及意义 3](#_Toc516516642)

[1.3 本文安排 3](#_Toc516516643)

[第二章 图像识别方案的简介 4](#_Toc516516644)

[2.1 计算机单目测距策略评估 4](#_Toc516516645)

[2.2 单目测距几何原理 5](#_Toc516516646)

[2.3 单目摄像头测距原理分析 6](#_Toc516516647)

[2.4 装置选择与搭建 7](#_Toc516516648)

[2.4.1 激光灯的选取 7](#_Toc516516649)

[2.4.2 摄像头的选取 7](#_Toc516516650)

[2.4.3 装置搭建 8](#_Toc516516651)

[2.5 本章小结 9](#_Toc516516652)

[第三章 机器视觉巡航与测距 10](#_Toc516516653)

[3.1 巡航方案设计与调整 10](#_Toc516516654)

[3.1.1 原始方案 10](#_Toc516516655)

[3.1.2 方案调整与工程重写 10](#_Toc516516656)

[3.2 图像识别测距过程与结果分析 12](#_Toc516516657)

[3.2.1 HSV颜色模型转换 13](#_Toc516516658)

[3.2.2 阈值分割 14](#_Toc516516659)

[3.2.3 高斯滤波 15](#_Toc516516660)

[3.2.4 Canny算子边缘检测 16](#_Toc516516661)

[3.2.5 计算质心与中心间距 17](#_Toc516516662)

[3.2.6 数据分析得出公式 17](#_Toc516516663)

[第四章 数据接收安卓App的设计实现 18](#_Toc516516664)

[4.1 数据传输方案的选择 18](#_Toc516516665)

[4.2 驾驶员端安卓App的开发实现 19](#_Toc516516666)

[4.2.1 传输内容定义 19](#_Toc516516667)

[4.2.2 传输格式定义 19](#_Toc516516668)

[4.2.3 安卓接收端开发： 19](#_Toc516516669)

[4.2.3.1 百度地图SDK 19](#_Toc516516670)

[4.2.3.2 界面展示与报警实现 20](#_Toc516516671)

[4.2.3.3 蓝牙模块 22](#_Toc516516672)

[第五章 数据无线传输策略 22](#_Toc516516673)

[5.1 数据传输策略的设计 22](#_Toc516516674)

[5.2 2.4GHz无线信号的收发 22](#_Toc516516675)

[5.2.1 Arduino和nRF24L01的连接方式 22](#_Toc516516676)

[5.2.2 2.4GHz信号收发的实现 23](#_Toc516516677)

[5.3 蓝牙信号的收发 24](#_Toc516516678)

[5.3.1 Arduino和蓝牙4.0模块的连接方式 24](#_Toc516516679)

[5.3.2 蓝牙信号发送端设计与实现 25](#_Toc516516680)

[5.3.3 安卓客户端蓝牙模块的实现 26](#_Toc516516681)

[5.4 无线数据传输策略评估 27](#_Toc516516682)

[第六章 总结与评估 28](#_Toc516516683)

[参考文献 28](#_Toc516516684)

[致谢 29](#_Toc516516685)

# 第一章 绪论

## 1.1 研究背景

### 1.1.1 我国当今道路交通安全现状

自上世纪80年代，尤其是改革开放以来，中国的科技取得突飞猛进的发展，汽车工业不断蓬勃奋进。自加入世贸组织以来,随着激烈的市场竞争及机动车尤其是汽车整车厂车辆制造水平的提高,车辆价格不断降低,车辆的社会保有量持续增加[1]。如今，汽车已逐渐成为居民日常生活不可缺少的交通工具。汽车，一定程度上可以代表人类现代工业文明。然而，汽车在给人类社会带来便利，推动社会发展的同时，也带来了一系列问题。交通事故是汽车给人类社会带来的影响最广，危害最大的问题之一。汽车由于灵活性高，数量大，驾驶员驾驶水平参差不齐，道路状况复杂多变，其交通事故率远高于火车、飞机等交通工具。为有效管理汽车交通，降低汽车道路交通事故的发生率，全球各个国家均提供巨大人力、物力出台交通法规，进行交通指挥和管控。经过多年努力，现如今欧美日韩等发达国家交通事故发生率趋于稳定。然而纵观国内，每年由于交通事故造成的经济损失依然巨大。同时随着汽车数量的不断增长，交通事故造成的危害程度还呈上升趋势。每年由于交通事故而导致的死亡人数依然居于世界首位。总体来看，我国的交通安全形势依然严峻。

高速公路的出现，大大提高了车辆长途行驶的速度，缩短了长途驾驶的时间。速度的大幅提升，推动了效率的迅速增长，对社会发展有明显的促进作用。另外，由于长途驾驶时间的缩短，减少了司机驾驶时间，一定程度上提升了安全性。然而，速度的提高意味着相同的反应时间内车辆的位移明显变大，一旦发生事故后果十分严重。同时又由于车辆的速度的明显提升，司机的视野范围有限，一旦前方出现交通事故，往往会由于躲闪不及时，造成二次事故和连环事故，据统计数据显示，高速公路上连环事故占有相当的比例。高速公路事故有翻车装车事故多、雨雾天事故多、停车事故多等特点[2]。如何让抛锚汽车或发生事故的汽车给予后方来车足够的预警和警示，已经成为一个十分重要的问题。

### 1.1.2 我国高速公路相关道路交通法规

三角警示牌是一个可以被动反光的三角形塑料制警示板。在高速公路上，如果驾驶员遇到车辆故障需要停车检修，或是遇到交通事故，可以利用警示牌的反光警示性，将警示牌放置在停车点后方一定距离的明显位置，从而对后方来车起到警示提醒避让的作用，避免二次事故的发生。

三角警示牌在避免连环事故发生上起到了十分重要的作用。交通法规规定告诉停车必须开警示灯，并在规定位置设警示牌。考虑到车辆速度、刹车距离的因素，在高速公路上出现故障，需要将警示标志放置在故障车来车方向150米以外。此项规定，可以最大限度地保证后方驾驶员可以注意到前方车辆的故障信息，提前采取行动，避免二次事故的发生。

### 1.1.3 图像识别技术与自动驾驶

计算机视觉的研究内容包括两方面 ,一是如何利用计算机实现部分人类视觉的功能，二是帮助理解人类视觉机理机器视觉技术[3]。计算机视觉是在 20 世纪 50 年代从统计模式识别开始的，当时的工作主要集中在二维图像分析和识别上[4]。机器视觉指的是使用图像摄取装置获取图像信息，通过图像处理系统使用多种处理算法进行图像的计算分析与处理，得到判别结果，进而根据此结果控制设备动作的一种人工智能领域的重要分支。通过机器代替人眼进行图像的识别、测量与判断，如果选用算法得当，处理系统完善，理论上可以获得比人眼处理更加智能、安全的效果。

自动驾驶汽车，是一种通过电脑系统控制的，理论上不需要借助借助驾驶员主动控制的智能汽车。目前应用在汽车上的防碰撞辅助驾驶主要用单目摄像头检测车道和前方车辆，雷达控制方向和速度并负责探测前方车辆、行人与障碍物[5]。在20世纪后半叶开始发展，进入21世纪后逐渐接近实用化。中国自动驾驶车在近十年也取得巨大发展，中国百度公司于2017年4月19日发布了名为“Apollo（阿波罗）”汽车自动驾驶软件开放平台，帮助搭建车辆自动驾驶系统。今年年初，Apollo无人车跨越了港珠澳大桥。互联网公司涉足自动驾驶领域，势必推动了自动驾驶技术的发展。

机器视觉技术、自动驾驶技术以及开放平台的出现与蓬勃发展，为本论文课题的研究提供了极大的帮助。在本课题车载机器人测量前方障碍物距离时用到了大量已经成熟的机器视觉技术，这些技术的开源，大大促进了本项目的进展。

### 1.1.4 安卓App开发现状

美国的谷歌公司与2007年正式公布了安卓操作系统 ，至今已有10年的发展历史，由于安卓开发采用Java语言，易上手易实现；安卓系统相较iOS有较大的开放性，早在2010年，官方认证的基于安卓系统的应用数已达超10万个。为便于安卓开发，谷歌公司在2013年推出Android开发环境Android Studio，该应用提供的集成开发环境，极大便利了开发和调试。

另考虑到安卓设备数巨大的时长份额，本项目中车载机器人与机主通信模块，接收端开发安卓App，用于接收有机器人发出的状态信号。

## 研究内容及意义

通过设计并改进基于机器视觉的车载安全机器人，旨在减少驾驶员行车过程中发生事故的隐患。众所周知，车辆行驶在高速公路的过程中，一旦发生车辆故障，必须依法停车排除故障。另外，如果发生了交通事故，车辆需要将车尽快移动到紧急停车区。无论任何一种情况，一旦车辆在高速上停车，必须依法在车后来车方向相距150米往上的位置处，放置三角警示牌。但是，必须考虑到从车辆停止到放置警示牌这段时间极高的危险性，人很难快速安全地将警示牌放置在正确的位置。

初代版本的护航机器人支持自动巡航与翻车警报，通过机器视觉的技术，可以完全脱离人工控制驶向既定的地点，停留到设定的时间后，自主掉头返回。然而，当前的三角警示牌并没有安全探测功能，并不能在有高速行驶车辆靠近甚至受撞击时，给出报警信号，从而提醒驾驶人员躲避。考虑到以上问题，综合考虑上版本车载机器人的不足。本项目给车载机器人增加测距测速模块，同时配备信号接收安卓设备，便于机器人检测前方来车的车距和车速的同时，可以在驾驶员端更新动态，在必要的时刻给予报警，最大限度地保障驾驶员和乘客的生命财产安全，避免二次事故的发生。

增加了测距测速模块以及同步状态客户端模块后，整个安全机器人的安全性和实用性可以有较大程度的提升。

# 第二章 需求分析与系统设计

## 2.1 需求分析

通过第一章的研究内容介绍，本工程的全部需求如下：

1. 实现车载安全机器人的自动巡航。
2. 实现基于机器视觉的车载安全机器人控制，实现机器人对目标物体的测距。
3. 研究并实现2.4GHz通信，编写安卓App实现智能车和安卓手机的通信。

## 2.2 系统设计

对照需求分析，设计本基于机器视觉的车载安全机器人的整体结构如图2-1所示：

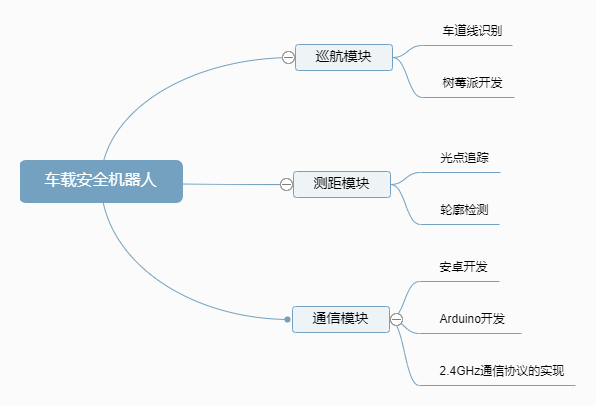


图2-1 系统结构设计

从此结构示意图可以看出，实现此机器人共需要完成三个大模块，对应三大需求：即基于机器视觉实现车道线识别控制车辆自动巡航需求，基于机器视觉实现激光点追踪和轮廓检测从而完成目标物体测速测距雪球，车辆安全状态和定位信息收发的通信需求。

本论文将对图2-1中指出的三大模块的实现过程分章节做详细阐述。其中第三章将重点阐述基于机器视觉的车道线识别算法的实现，并根据识别结果使用树莓派串口控制小车的运动；第四章将重点介绍基于机器视觉的光点检测测距算法的实现；第五章将重点阐述从智能车到安卓手机的整个通信过程的实现方法，分别介绍安卓App开发过程、Arduino模块的开发过程以及2.4GHz通信的实现过程。

三个模块协同配合，实现整个系统的完整运行。

# 第三章 图像识别方案的实现

## 3.1 计算机单目测距策略评估

计算机视觉的基本任务之一是从摄像机获取的图像信息出发计算三维空间中物体的几何信息[6]，从一定程度上讲是模拟人眼的图像处理过程。人之所以可以看到图像，首先是双目感光在视网膜上获取图像，通过人脑对图像信息进行计算处理，从而使人能感知到物体的大小，距离人眼的距离甚至移动速度。模拟此过程，最单纯直接的想法便是使用双摄像头对获取图像的视差进行分析，得到图像的位置信息。然而双目测距对处理器有较高的要求，且实现起来难度较大，并不适合在本项目中使用。于是本项目考虑单目测距策略。

实验表明人在盖住一只眼睛的情况下，感知视线内物体的距离是比较困难的。因为单目仅能获取一幅二维的图像，看到的图像处于同一个平面，距离信息并不能直观的获取。这一理论知道了手机厂商对摄像头的选择，为了使手机拍出类似广角镜头的效果，当今主流厂商选择了给手机配备双摄像头，拍出广角效果的同时，可以给背景增加虚化效果，从而使物体的距离信息清晰可见。

然而，单目并非不能得到距离信息。但通过限制图像中的Y轴坐标，可以获得物体的位置信息，如图3-1所示。

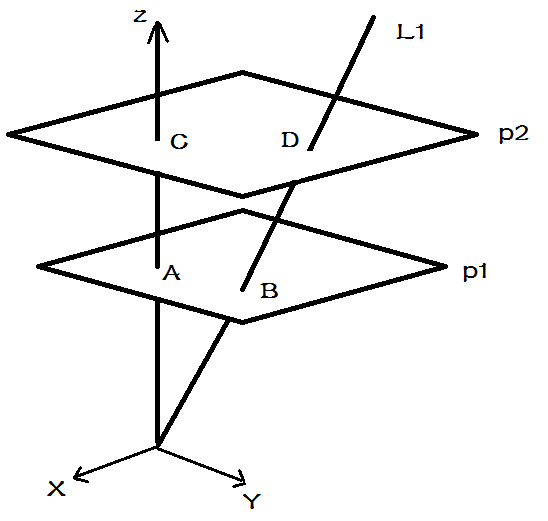


图3-1 单目测距原理简析

如图所示，在三维空间xyz中，存在坐标轴圆点距离不同的两个平面P1和P2，与z轴分别相交于点A和点C。如果在空间中发出一条与z轴不平行的直线L1，则L1和两个平面分别相交于点B和点D。则线段AB和线段CD的长度即可反映出两个平面和圆点的距离。显然平面中两交点的间距越长，平面离圆点的距离越远。

通过以上分析，单目测距在理论上是可行的，只要可以找到这两条参照线，经过适当的计算，可以求取任何摄像头可见物体的位置信息。

## 3.2 单目测距几何原理

从3.1的分析中，可以发现，只要找到图3-1中所示两条直线，通过对图像中的距离信息进行一定转化，就能将图像中的距离信息转化为物体到摄像头的距离信息。于是问题的关键在于如何选取这两条直线以及如何进行转化。

通过分析可以发现，并非任意两条直线都能满足测距要求。经过尝试和思考，除了两条平行线无法满足要求外，如果两条直线均可移动，则通过测量两直线在图像中的交点间距便不具有测量意义。因此，必须保证一条直线处于固定状态。

为便于理解，在这里将图3-1中的坐标轴原点当作想要测算距离的物体表面的某一个点，则物体到达摄像头的距离可以等效为原点到平面的距离。此时不妨令平面P1平行于平面P2，且均垂直于z轴，则不难做出同时过z轴和直线L1的截面如图3-2所示：

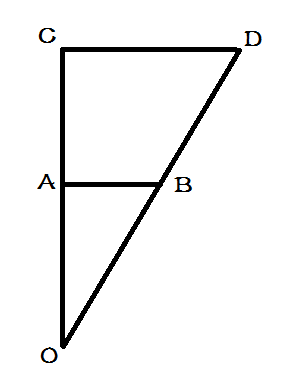


图3-2 单目测距截面图

在图3-2中，根据三角关系，不难得到三角形OAB相似于三角形OCD，则存在比例关系：

（式3-1）

通过对比例关系的数学变换，可以得到：

（式3-2）

在该式中，由于AB属于参考平面P1，可以认为是固定平面，因此AB和OA的长度认为是定值，因而 是一个比例系数，待测平面与观测点的距离OC与CD的长度呈线性关系。理论上来看，CD是可测量距离，为求取OC的长度，问题的关键在于如何求取比例系数。

通过分析可见，本项目需要测算的是水平方向上的距离AB和CD。只有实现约定选中某一固定点测算才有意义。通过比较分析，最适合作为固定参考点的应当是整个画面的正中央。由于本项目已经成功得到了水平偏移距离，比例系数其实可以通过实验获得。在保证参考平面P1始终与z轴垂直的基础上，通过多次改变P1的位置，多次测量OA和AB的距离，算出比例值求取平均值，从而可以较为准确确定装置对应的比例系数。一旦比例确认后，无论物体在任何位置，只要在图像上可以计算出CD距离，都可以直接求取物体离装置的距离。

## 3.3 单目摄像头测距原理分析

从3.2分析可知如果想要获取比例系数必须先能获取CD的距离。由于直接测算CD的距离没有意义，毕竟如果可以测算CD距离，直接测量物体到观测点的距离也不是难事。虽然不能直接测算出CD，但可以读出线段CD在摄像头所成的像中对应的长度，根据二者之间存在的比例关系即可便捷测算出CD长度，进而获取测距信息。

相机的镜头是一面凸透镜，根据凸透镜成像原理，从无限远处射来的光线，可以近似当作是平行于光轴的光线，将会通过透镜焦点。如果光线发射自有限的距离，经过透镜的折射后，并不会通过焦点，从而会在焦点处的平面上与中心有一定距离。这其实类似人眼成像原理，此平面就是人眼的视网膜。从下图3-3中可以更清晰地看到这一过程：

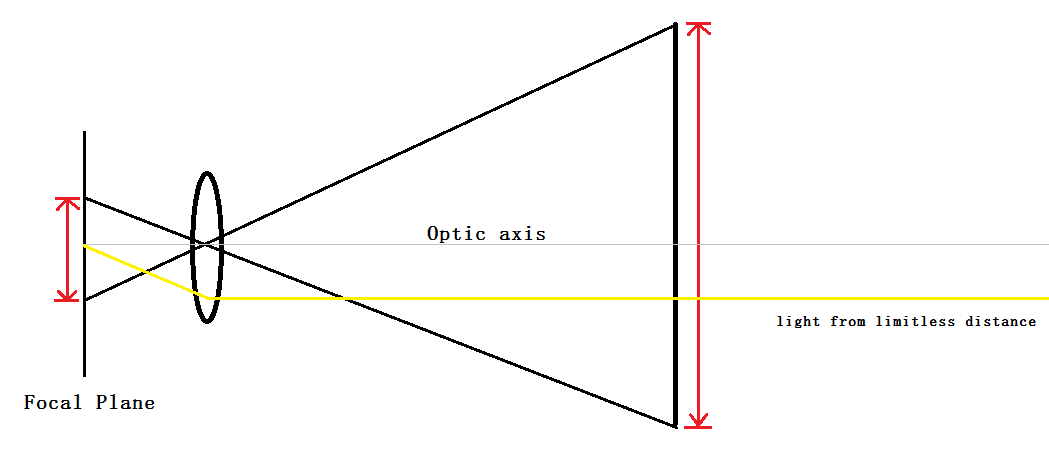


图3-3 凸透镜原理参考图

图3-3中，黄色的光来自无限远处，经过透镜折射经过了焦平面的正中间，也就是图像正中央的位置，而在距离透镜有限距离处的物体反射的光线经过透镜折射后成像在了偏离中心的位置，在这里可以利用此偏移距离利用三角关系来测算物距。

## 3.4 装置选择与搭建

### 3.4.1 激光灯的选取

在3.2的讨论中可知，如果想要利用好相似三角形中的比例关系，则必须找到可以和透镜水平方向距离固定的物体，同时这个物体可以在被测物体上找到映射。通过这一分析，很自然地想到了使用激光灯。选用激光灯的优点很明显：首先，激光灯可以发出强度较高的光线，在摄像头中很容易可以捕获，通过对图像进行简单的阈值分割即可捕捉到光点，进而便于进行后续的分析；其次，激光灯可以发出不易散射的光，只要能量足够，可以保证百米距离也可汇聚成点；另外，激光灯体积较小，且方便使用管脚控制亮灭，非常适宜嵌入本项目的车载机器人中去。



图3-4 测距装置选用5v激光灯

经过调研，本项目最终选定了5V供电的小型激光灯如图3-4所示，这款激光灯的优点在于其在保证光强的同时做到了体积尽可能小，同时供电方式简单。考虑到车载机器人的便携性和成本，此款激光灯最符合要求。

该激光灯的投射方向需要与摄像头拍摄方向一致，且必须固定在摄像头旁边，保证与摄像头的相对位置适中不变。

### 3.4.2 摄像头的选取

理论上说摄像头的分辨率和采样频率越高，越利于图像的识别与处理。但同时必须考虑到成本的问题以及必要性。鉴于本项目选用了光强较高的激光灯作为识别目标，另结合图像识别中的阈值分割技术，不需对摄像头的成像效果要求太高即可达到初步的处理目标。

经过调研和测试，摄像头可以做到每秒采样帧数在25到35以上，分辨率在640×480以上即可满足需要。为了是整个装置足够简单，特意选择了免驱动摄像头，如图3-5所示，此摄像头与控制端之间使用USB进行连接。



图3-5 测距装置选用USB摄像头

### 3.4.3 装置搭建

根据以上3.3中对此车载安全机器人单目摄像头测距原理的分析，整个装置设计为如图3-6所示。

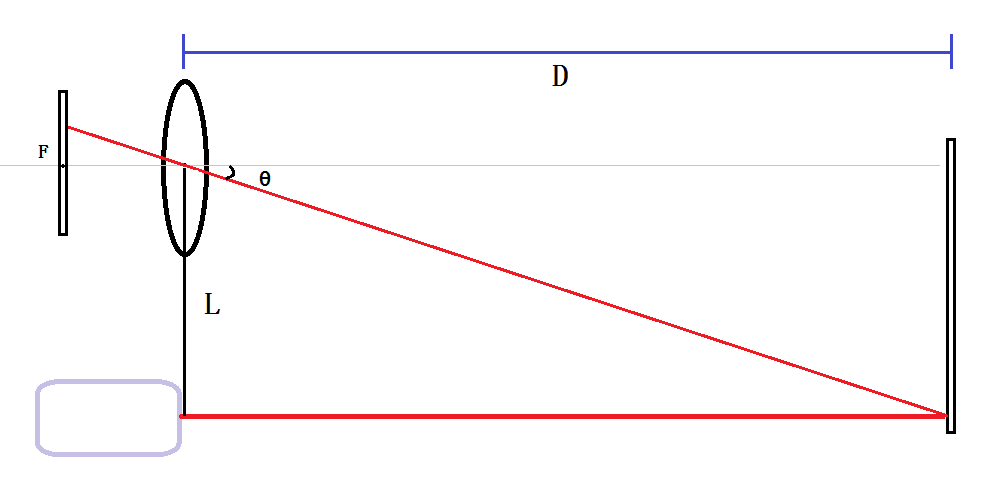


图3-6 测距装置完整示意图

图3-6中，假设待测物体与摄像头距离为D，激光灯固定在与摄像头光心水平距离为L的位置处，则根据图中的三角关系不难得到：

D = 式（3-3）

显然，L是已知可测距离，问题的关键是确认每一个距离对应的θ角。根据上图中对等角关系，在摄像头的像空间上，弧度θ其实就等于图像中光点所在位置距离图像正中央水平偏移像素点个数num与每个像素点对应的光心弧度radian，再加上误差值offset，即是：

θ = num × radian + offset 式（3-4）

从直观上进行分析，不妨假设被测物体距离摄像头无穷远，假使激光灯发出的激光不会衰减，则打在物体表面的光点反射回来的光线可以近似认为与光轴平行，则根据凸透镜成像原理，平行于光轴的光线经过透镜折射后将会通过焦点。即图3-6中点F，此时θ角为0度，摄像头获取的图像中，光点的水平坐标就在中央。当物体不断靠近摄像头，光点的位置也会逐渐偏离中心，当物体无限接近于摄像头，θ角将趋于π/2。

综合3-3式和3-4式，可以得到完整距离公式如下：

D = 式（3-5）

公式3-5中需要确认的未知数有每个像素点对应的光心弧度radian，以及与设备相关的误差offset。两个未知数，理论上只需两个方程即可确认，但由于实验次数较少误差较大，设计为多次实验排除错误数据后，求取两个参数的平均值，得到准确的距离公式，用于基于机器视觉的车载安全机器人的测距。

## 3.5 本章小结

本章从测距理论角度，循序渐进，从简单到具体阐述了单目测距的几何原理、摄像头成像与测距原理以及本项目测距装置的选取和搭建。实际上本章中三次阐述的测距原理从理论上是统一的。总结来看，其实用到的是一种等效思想，使用适当的数学知识，将实际问题进行转化。用最简单的想法去考虑，本单目测距理论可以类比人眼识物远小近大的原理。通过类比思考，数学几何分析，结合手中的硬件，通过实验探索与调研，得出本测距的思路与努力方向。通过本章的探究，明晰下一步探究的方向，即通过多次实验得出与装置相关的两个参数。但由于原车载机器人平台的种种原因，需要先将整个机器人平台转至树莓派。下章将从平台转移和图像识别分析并得出最终公式两部分来介绍。

# 第四章 机器视觉巡航与测距

## 4.1 巡航方案设计与调整

### 4.1.1 原始方案

本车载安全机器人初版demo是基于Intel处理器的，但由于Intel处理器功耗较高，且接口较少，拓展性不佳且成本较高，最终决定将原工程迁移至其他更合适的平台上。

树莓派具有低功耗，高性能，价格适中，接口多且全，可拓展性强等优点。为了实现本工程的全部要求，本项目选用了支持WiFi的树莓派3 model B（如图4-1所示）。

初版车载安全机器人的中控板选用了Minnowboard板，原工程完全基于Linux操作系统，对底层硬件的依赖很少，且整个主板仅外接了4个USB接口。



图4-1 本工程选用树莓派

### 4.1.2 方案调整与工程重写

### 4.1.2.1 安装系统，打开ssh和vnc服务

树莓派从零开始，安装官方镜像系统后，为了便于远程控制，打开了ssh和vnc服务，树莓派3的优点在于支持WiFi，通过给树莓派外接显示器，打开两个服务十分方便。

在开启vnc服务后，通过在Terminal中输入hostname –I查看本机IP地址后，即可在同局域网中其他Windows设备中图形化登录树莓派。在任意端对同一树莓派的操作会发生叠加。通过vnc服务，可以十分方便地进行协同工作。

### 4.1.2.2 配置OpenCV环境

在树莓派中配置OpenCV的过程是相对比较复杂的，通过查阅各种书籍资料，并向其他同学请教讨论，给出配置过程如下所示:

首先执行的是更新操作，更新apt-get以及rpi，安装OpenCV的构建相关命令。在执行此步骤时，一开始会遇到下载非常缓慢的问题，这是因为软件源设在英国，下载速度十分缓慢，只需执行指令将软件源切换到国内镜像（本项目中选用了中科大的镜像），执行速度会出现巨大的飞跃。更新后，在树莓派上按顺序安装了OpenCV的构建指令、图片工具包、视频I/O包、GTK以及优化函数包，通过github直接下载并安装OpenCV3.0和opencv\_contrib，安装完python开发包和pip等工具后，项目创建出虚拟工作环境，在此环境下安装numpy，编译安装并配置opencv3.0，随即成功配置好OpenCV环境。

### 4.1.2.3 配置Qt环境

Ubuntu平台配置Qt环境相对比较容易，由于之前将软件源切换至了中科大，通过命令行直接下载安装好Qtcreator，配置好opencv的相关库后即可在Qt环境下完成相关机器视觉工程的开发。

### 4.1.2.4 道路线识别和巡航算法实现

由于项目的demo在Intel芯片上已经实现。在一开始迁移代码时，经过分析我认为由于原工程建立在Intel芯片上的Qt环境中，对硬件底层的依赖并不多，因此理论上只需将原工程的代码在新平台上调通即可。然而在实际操作的工程中，遇到了不少问题。

首先，经过分析发现，由于原工程完成在三年前，其中多数opencv库函数已经有了更新，因此在配置依赖的时候，为了让整个工程和依赖匹配，需要更改的东西很多，工作量远超预期。

其次，即便完全调通原代码，由于新需求对摄像头的位置十分敏感，原工程并不能满足新需求，需要做的调试会增加很多。

综上，我最终决定不再迁移原工程，而在新平台上重写整个工程。受限于篇幅，在这里只对整个工程的完成过程做简单介绍。原需求需要完成车载机器人通过前方图像的获取，分析图像提取出道路线信息，再通过对道路线信息的处理，控制智能车调整速度、方向，从而完成整个车辆的自动巡航。

基于特征来识别道路线的方法主要是利用道路线的边缘、纹理、颜色、频域特征等来对车道线进行识别[7]。基于视觉的车道线识别对于道路周围环境的光照、天气较为敏感，需要对获取的照片先做预处理才能获得更好的识别效果[8]。车载摄像头获取到前方图像信息后，处理器先根据测试结果对图像进行尺寸截取，从而减少天空对道路线识别的影响。其次，对图像进行灰度化处理，过滤掉色彩对图像处理的影响。通过对称阈值切割和膨胀腐蚀处理，捕捉图像最突出的信息。然后使用canny算子边缘检测技术，通过调整参数，即可将道路线捕捉出来（图4-2中的红线）：



图4-2 经过canny算子边缘检测后识别出的道路线

由于图像中的道路线可能有多条，需要程序自己通过加权提取中真正用于控制使用的道路线。在本工程中我通过捕捉直线的特征，提取出最内侧的两个直线而将其他直线过滤掉。

提取到参考道路线后，接下来我将通过分析整个图像中心点偏离两道路线中心点的距离，判断车辆运行的方位偏移，并根据此方位信息，给车辆发送转向信号和速度控制信号，经过多次实地调试，调整车辆速度和转弯方向，最终维持智能车相对稳定地直行到既定的位置。

## 4.2 图像识别测距过程与结果分析

通过4.1中的重写工程，本项目在树莓派上实现了智能车的自动巡航，保证智能车可以停在既定的位置，为下一步的测距建立基础。本节将正式阐述基于机器视觉的距离检测技术在树莓派上的实现过程。

整个车载安全机器人的视觉算法将在树莓派上的qt环境中开发完成。全部定位并识别的历程涉及到hsv色彩模子的转换，根据调试出来的算法找到分割阈值，利用此阈值对全部图象进行分割，对过滤后的图象使用高斯滤波，再对滤波后的图象利用Canny算子边沿检测算法找到图象中的激光点，确认其精确位置。最后通过质心坐标公式得到激光点质心坐标，从而得到照射在待测物体上的激光点水平偏移像素点个数num。通过实地测量待测物体离摄像头的距离D。以上整个过程称为一次测量，一旦得到一组数据，带入到3-5的公式中即可得到一个方程式。通过联立两个方程式即可求得一组radian和offset数据。通过多次测量，求得radian和offset 的平均值即作为本车载安全机器人的测距公式。

### 4.2.1 HSV颜色模型转换

计算机视觉领域的很多处理算法都是基于RGB颜色空间模型，RGB模型是用三原色红绿蓝色光进行混合形成各种色彩，这与人类的双眼的感光生理特性有关，因而这种色彩模型非常容易被理解。然而在RGB模型中，三个维度无直接联系，不能表明色彩之间的关系。HSV模型又称为倒锥形模型，其中H表示色相，S表示对比度，可以反映出出像素点的色彩深浅，随着S值的变化，图像会从纯灰度向色彩斑斓转化；V体现的是像素点是否明亮，V值越大，图像越明亮。RGB模型和HSV模型如图4-3和图4-4所示：

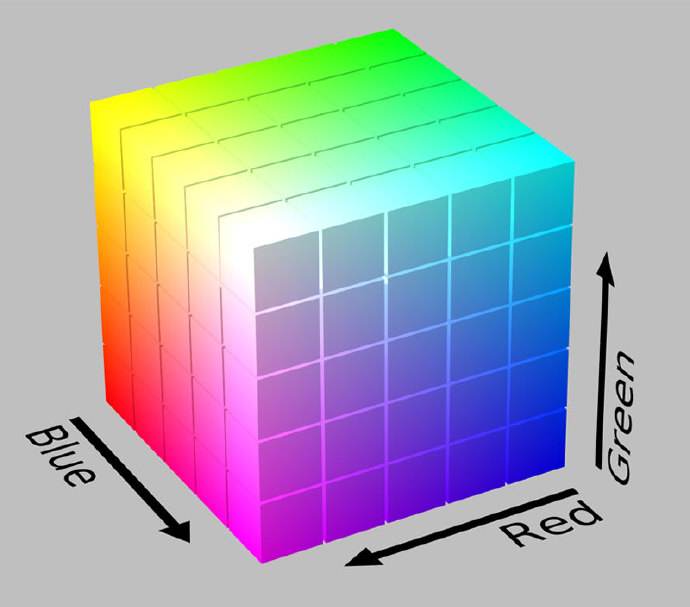


图4-3 RGB模型

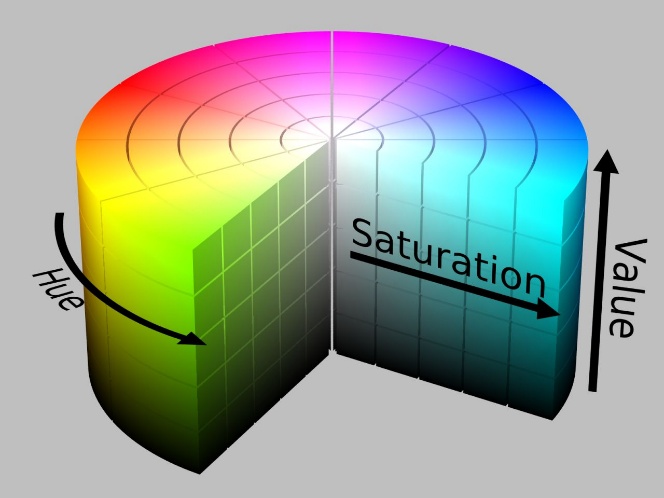


图4-4 HSV模型

从两个模型的色彩分布直观来看，HSV模型中，相近的色彩分布更为集中，同时由于HSV中三个维度反映的色彩信息有直接的联系，本项目最终选择使用HSV模型来进行阈值分割。由于H和S的值代表了色彩信息，因此理论上，通过确定阈值，限定H和S的值，即可对整个图像进行阈值分割。

### 4.2.2 阈值分割

单色图片和彩色图片相比，数据量的差距是巨大的。通过设定特征阈值，将一张图片中的各个像素点划分成若干类，最大程度地压缩图像的数据量，同时简化后续的数据处理与图像分析。因此，在实现此车载安全机器人近一半识别算法前，先确定阈值对图像进行分割。

在车载机器人整个行进过程中，不妨提取一帧图像进行阈值分割的过程分析：

首先，将图像以IplImage的格式读入，使用OpenCV 库提供的转换函数cvCvtColor对整个图像模型转化。从而使整个图像从RGB模型，转化为HSV模型，即使用：

cvCvtColor(mySrcImg, myHsvResult, CV\_BGR2HSV)；

使用cvGet2D函数获取转化后的myHsvResult图像中每一个像素点的hsv值，通过比较每个像素点的Hue（色相）值，将满足筛选范围的像素点的hsv像素值保留，不满足筛选范围的像素点置为黑色，即可实现阈值分割。

在探究特定色彩的HSV值分布时，查阅资料得到了以下的对照表，相关研究人员通过实验得出了常见色彩的HSV范围值：

表4-1 常见颜色HSV值对照表



通过查询上表结合实验，可得激光灯对应的红色的HSV范围，即 Hue值的范围在0~10和156~180之间。以此范围来过滤整个图形，处理过程为：

CvScalar s;

IF ((scalar.val[0] > 0 && scalar.val[0] < 10) || (scalar.val[0] > 156 && scalar.val[0] < 180))

THEN s.val[0] = 0, s.val[1] = 0, s.val[2] = 255

ELSE s.val[0] = 0, s.val[1] = 0, s.val[2] = 0

图像上只剩下了红色光点，这为接下来确认激光点的位置，并计算质心提供了先决条件。实验截图如图3-5所示：



图4-5 滤波前

可以看到激光灯打到的区域可以被清晰地定位出来。但由于图像的像素限制，红点边缘的噪声很多，很难看到清晰或者集中的光点区域。为了过滤噪声，降低细节层次，增强对比度，结合查阅资料，最终决定在进行阈值分割之前使用高斯滤波来降噪。

### 4.2.3 高斯滤波

高斯滤波是一种图像过滤技术，其脉冲响应特性曲线是一个高斯函数。可用于消除高斯噪声，在本车载安全机器人的图像处理工程中，由于处理需求对图像的清晰度要求较低，因此可以先对图像使用高斯滤波降噪。图像经过滤波降噪算法后，其细节信息被隐藏，处理的难度明显降低。

OpenCV库为高斯滤波的实现直接提供了封装好的函数，其含有6个参数。其中前两个参数src和dst是输入图像和输出图像，此方法对本项目机器人输入的图片深度有限制，此车载安全机器人自带摄像机获取的图片格式为CV\_8U；之后的ksize变量表示的是高斯内核的大小，ksize的宽度和高度可以不同，但必须为正数，本工程选择的内核大小为Size(7, 7)；再接下来的两个double型变量表示的是在X和Y方向上的高斯滤波函数的标准偏差。

通过配置参数，选择sigmaX和sigmaY都等于100，经过高斯滤波后，图像的变化如图4-6所示：



图4-6 滤波前后对比

从图4-6中可以发现，激光点所在的区域已经比较明显了，整个画面的信息量得到很大程度的减少。由于程序选择的阈值合适，图像中的干扰点较少，在定位了光点之后，接下来的工作主要在如何找到光点的质心，从而确认其和画面中心的水平偏移像素。

### 4.2.4 Canny算子边缘检测

求取图像上某一图形质心的前提是找到此图形的轮廓。本项目在4.2.3节中已经成功将待测激光点定位出来，需要执行算法将光点位置的轮廓获取到。在opencv2的库函数中提供了canny函数的接口。Canny在当年的研究中提出了尽可能标识实际边缘、标识的边缘要尽可能接近实际边缘以及图像噪声不应标记三个边缘检测的目标。显然在整个处理过程中，会经历去除噪声，根据亮度梯度定位边缘的步骤。实际上所有边缘检测算法利用的都是图形边缘线两侧灰度级的突变特性

canny函数一共有6个参数。前两项分别为输入的灰度图像和输出图像，后面接续的两个double类型变量表示灰度值的最小阈值和最大阈值，通过判断输入的灰度图像中每一个像素点是否在阈值范围内，来确认像素点是否为边缘，通过标记可以检测出对应阈值范围内的整个图像的边缘。

经过以上过程的处理后，想要进一步获取本车载安全机器人图像的轮廓信息，可利用库函数findContours，此函数可以使用向量保存图像边界连续的像素序列，通过筛选，将向量中含有的所有轮廓序列标示出来，从而不难获得图像的轮廓，用于进一步分析研究。

findContours函数共有五个参数。第一个参数一般选取的是经过canny算子边缘检测后二值的图像；第二三两个参数是本工程要事先定义好的轮廓向量组以及层次向量，需要注意的是第二个参数本身是一个vector，同时其每一个元素又是一个vector，用来存储的正是上文所说的像素序列，每一个序列表示一个轮廓；第四个参数定义的是轮廓的检测模式，取指有四，通过不同取值，可以获取不同的轮廓信息。通过分析比较，本工程初步只需将最外层轮廓检验出来即可，如果噪声干扰去除不理想，再去考虑调整轮廓检测模式；第五个参数定义的是轮廓的近似方法，通过多次实验尝试的对比，本工程最终选择使用CV\_CHAIN\_APPROX\_SIMPLE的模式，即仅保留轮廓的拐点信息，而忽略掉拐点之间的线的数据。

通过调用findContours函数，可以得到整个图像中所有的轮廓信息并将它们保存在了contours向量当中，供后续处理。

### 4.2.5 计算质心与中心间距

由于已经已经获取了图像的所有轮廓，为了便于操作这些轮廓，本项目引入了轮廓矩（Moments）的概念，OpenCV提供了便捷的moments()轮廓矩计算函数，原型如下：

Moments moments(InputArray array, bool binaryImage=false )

只需输入所有的轮廓信息并将第二个参数置为false，即可计算出整个图像的所有矩。每一个Moments对象，具有空间矩，中心距和中心归一化矩。通过查阅资料，使用图像的零阶矩和一阶矩可以计算出形状的质心，质心坐标公式如下：

式（4-1）

通过此方式计算质心的优势在于，即便有外部噪声的干扰，获得的质心并不会有明显的偏离。通过此方式可获取到图像的轮廓，即可得激光点的质心坐标，接下来只需做差即可得到激光点到图像正中心的距离：

unsigned int pixFromCentre = mySrcImg.cols / 2 - mc[i].x;

### 4.2.6 数据分析得出公式

重新梳理下从4.2.1到4.2.5整个过程，通过摄像头获取一帧图像，经过高斯滤波并在HSV模型下阈值分割与过滤，之后通过canny算子边缘检测并获取激光点的轮廓数据，参考公式获取了激光点质心坐标，通过作差得到了质心的水平偏移，即4.2章开头时提到的待求数据num，此时测出待测物体距离摄像头的真实距离D和以及摄像头和激光灯的垂直于光路方向的距离L，这里L为1m即得到一个关于误差offset和每个像素点对应的光心弧度radian的二元一次方程。保持L的值不变，调整待测物和摄像头的距离D重新测量，得到另一方程，联立成组，解得offset和radian。重复以上的过程，对解得的两个参数求取平均值。下表展示的是本项目10次实验结果：

表4-2 实验结果

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | D | num | 方程式 |
| 1 | 3 | 224 | 1 = 3 \* tan( \*radium + offset) |
| 2 | 6 | 112 | 6 = 3 \* tan( \*radium + offset) |
| 3 | 9 | 73 | 7 = 3 \* tan( \*radium + offset) |
| 4 | 10 | 65 | 10 = 3 \* tan( \*radium + offset) |
| 5 | 12 | 53 | 12 = 3 \* tan( \*radium + offset) |
| 6 | 13 | 49 | 13 = 3 \* tan( \*radium + offset) |
| 7 | 14 | 45 | 14 = 3 \* tan( \*radium + offset) |
| 8 | 15 | 42 | 15 = 3 \* tan( \*radium + offset) |
| 9 | 16 | 39 | 16 = 3 \* tan( \*radium + offset) |
| 10 | 18 | 34 | 18 = 3 \* tan( \*radium + offset) |

保持L的数据不变，通过更改D的距离，前后得到10组数据，联立解得5组offset和radian的值，求取平均值后确认在L=1m的情况下本车载安全机器人的距离公式为：

D = 式（4-2）

此公式中仅有一个通过摄像头可以直接获取的变量num，因此只要将此程序设为开机自启动，树莓派端可以实时获取前方障碍物的距离。至此，本基于机器视觉的车载安全机器人的测距模块即介绍完毕。

# 第五章 数据接收安卓App的设计实现

## 5.1 数据传输方案的选择

已经驶离驾驶员150米的机器人已经获取了其四周车辆的距离信息，但如何将此信息传回驾驶员处是本项目要解决的另一问题。

为实现从车载机器人到驾驶员安卓客户端的信息传输，需要选用最为合适的通信协议，既要保证通信过程可以保质保量，又要尽可能降低成本。由于本科阶段自学了安卓的开发，同时做过socket通信，项目开始阶段首先尝试使用基于TCP的socket进行通信，因为本科阶段有过类似的开发实践，使用此策略开发成本较小，但考虑到使用socket通信需要使用网络，200米的露天范围内不方便建立局域网，必须在公网中申请一个公共的服务器进行数据包的转发，整个过程还是相对比较复杂，考虑选用其他的通信方式；考虑到树莓派自带蓝牙模块，且安卓手机都支持蓝牙的检测和配对，开发比较容易，但由于蓝牙的传输距离太小，无法达到本项目要求的150米的距离，因此考虑放弃此策略。2.4G无线通信技术有着适用范围广、带宽高、耗电低等特点[9]。综合考虑通信的距离可达性以及通信模块的成本，开发难度等因素后，本项目最终选用2.4G无线通信，经过查阅资料此通信协议可以支持150~200米甚至更远距离，且安卓手机支持2.4G网络的检测和连接，综合这些特性，本项目最终选定了2.4G无线通信协议。

## 5.2 驾驶员端安卓App的开发实现

### 5.2.1 传输内容定义

在论文第一部分最终获取到了车载安全机器人前方车辆的距离信息，距离是智能车要传输的第一个重要信息，安卓客户端获取后可对距离信息进行处理，当满足一定的条件后在客户端对驾驶员进行报警；其次，由于初版的车载机器人配有GPS模块，可以获取其位置信息，可通过2.4G通信协议将位置信息发送给驾驶员，让驾驶员实时掌握车载机器人的位置。

### 5.2.2 传输格式定义

将1中定义的距离信息和位置信息包装成json格式进行传输。样例包如下：{state : ‘safe’, GPSlong : ‘37.96’, GPSla : ‘116.36’ }

### 5.2.3 安卓接收端开发

本人于大三利用课余时间学习了安卓的基本开发，了解并基本掌握了安卓的基础知识，可以自行设计和开发简单App，本项目中我通过学习百度LBS开放平台提供的地图SDK，将车载机器人的实时位置显示在驾驶员客户端上。通过自学2.4G通信协议，建立起了小车和手机App的通信，本节将就几个重点模块的设计开发做基本介绍。

### 5.2.3.1 百度地图SDK

感谢百度公司提供给开发者的LBS开放平台，为本项目功能的实现提供了极大的便利。通过登录开发平台下载百度地图的基础地图和定位功能两个SDK，解压后将得到的libs目录中的每一个文件直接添加到新建的安卓项目中的正确位置。

谷歌公司为开发者提供了Android Studio这一集成开发平台，使得安卓项目的开发变得十分方便。通过新建项目，在项目目录的app模块下方找到了libs目录，将从开放平台中下载的BaiduLBS\_Android.jar 文件放在此目录下；之后在src/main目录中新建名为jniLibs的文件夹，存放从开放平台中下载的所有so文件，添加后的目录如图5-1所示；最后是更新gradle文件，添加Jar后缀的包到安卓项目的libs目录中后，通过同步建立起工程对Jar包中接口的引用。完成以上的步骤后，来自LBS开放平台的SDK即就绪，通过后续代码的调用可以直接使用百度地图的接口来进行定位并显示位置。

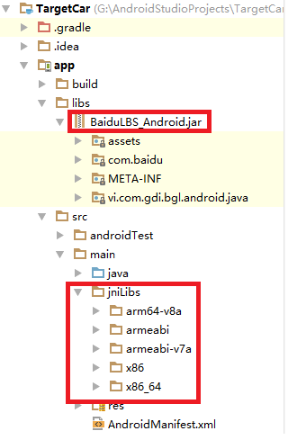


图5-1 添加SDK后的目录截图

### 5.2.3.2 界面展示与报警实现

通过参考整个车载机器人的需求分析，驾驶员App的界面初步设计为下图所示的格式，首先在最上方放置一个TextView部件显示当前小车的经纬度；页面的主体会显示地图，通过调用百度提供的接口显示出类似百度地图的效果，图上标注出小车的实时位置；在地图下方的明显位置给出小车周围最近车辆的距离，当距离进入一定的阈值会在此位置给出明显的报警信息，并使手机持续震动发出报警信号。



图5-2 Android端界面显示

首先是经纬度的确定，需要先在AndroidManifest.xml文件中为百度LBS SDK申请所有用到的权限，并增加<meta-data>声明LBS SDK服务并填入从百度申请到的API Key。在实现经纬度获取的时候，项目创建了LocationClient实例并注册了一个定位监听器。由于在配置文件中添加了包括获取位置信息等运行时权限，因此在运行本程序时，需要首先征求用户的权限。权限获取后，获取定位的方式很简单，只需调用创建的LocationClient的start()方法即可，结果会回调监听器，这里只需使用监听器的onReceiveLocation()方法中getLatitude()和getLongitude()函数，即可获取当前的位置的经纬度，通过实现设计好的格式组成一个字符串，直接显示到TextView上。

为了让小车的位置可以直接显示在地图上，只需在主活动的xml文件（activity\_main.xml）中增加地图视图的模块即可，将完整包名为com.baidu.mapapi.map.MapView的控件放在界面的主体位置即可，修改主活动初始化MapView对象，并将其显示出来即可。为了在地图上标注出小车的位置，百度LBS SDK为开发者提供了MyLocationData.Build类，只需新建builder对象，将通过2.4G通信模块传入的小车的实时经纬坐标作为参数传入函数，调用builder的build()方法，并将MyLocationData实例传入到setMyLocationData()方法中去，小车的实时位置即可在地图中显示出来。

当驾驶员端收到了小车周围车辆靠近的距离信息后，只需将此字段提取，显示在页面下方即可，一旦距离小于20米，手机将持续震动。震动则通过Notification的API中的setVibrate()方法实现，将参数设定为new long[] {0, 1000, 0, 1000}后，即可实现安卓手机的持续震动。在使用震动之前，App震动权限的声明是必须要完成的。当客户端收到的距离信息触发了震动的条件，手机将持续震动为驾驶员报警，由于驾驶员和小车还有150的距离，因此驾驶员有相对充足的时间进行躲避。

### 5.2.3.3 蓝牙模块

在接下来的第六章将详细介绍整个无线数据传输策略，在驾驶员安卓App端将会使用蓝牙接收最终传来的无线信号并展示。

# 第六章 数据无线传输策略

## 6.1 数据传输策略的设计

近年来，无线通信技术快速发展，涌现出蓝牙，Zig-Bee，WiFi和RFID等无线通信技术[10]。项目最开始计划使用蓝牙来完成无线数据的传输，因为安卓端自带蓝牙接收模块，实现起来比较简便。但考虑到150米以上的通信距离，考虑放弃蓝牙而选用普通2.4GHz的无线通信，并选用业界常用的2.4GHz信号发射模块nRF24L01+PA+LNA已保证150米以上的传输效果。在研究的过程中，调研发现安卓对2.4GHz无线信号并没有做特殊的模块支持，如果想要尝试无线信号的直接收发处理，需要修改传输协议，实现起来需要花费更多的时间成本。经过进一步的技术调研和评估比较，我决定采用以下方案：

保留蓝牙通信和2.4GHz通信，使用选定的arduino模块进行信号中转。车载安全机器人端将数据打包好后，通过nRF24L01模块发送出来，被在150米外的中转端的另一个nRF24L01模块接收到，信号立即通过中转端的蓝牙模块发送出去给安卓手机，安卓客户端接收到数据包后对数据进行解析，展示在App上，整个数据通信完成。

综上，本项目需要增加一个arduino单片机、2块nRF24L01+PA+LNA模块以及一个蓝牙4.0模块。

## 6.2 2.4GHz无线信号的收发

### 6.2.1 Arduino和nRF24L01的连接方式

使用杜邦线按照下表中的指示连接Arduino UNO模块和nRF24L01：

表6-1模块连接方式

|  |  |
| --- | --- |
| Arduino UNO | nRF24L01 |
| 3.3V | VCC |
| GND | GND |
| 7 | CSN |
| 8 | CE |
| 11 | MOSI |
| 12 | MISO |
| 13 | SCK |

连接成功后的信号中转模块如图6-1 所示：  


图6-1 连接nRF24L01模块

### 6.2.2 2.4GHz信号收发的实现

通过查阅资料， Arduino官方提供了开发nRF24L01模块可用的开源API，在其官方网站<http://playground.arduino.cc/InterfacingWithHardware/Nrf24L01>上，可以找到Mirf模块，通过下载解压，将其放置在Arduino安装文件中的libraries中，便可以在之后的开发中调用此API。除了用到此API，需要的头文件还有SPI.h。

在数据发送端Arduino的setup()函数，可以直接调用官方提供的Mirf中的初始化函数，使用5个字符的长度定义好发送端地址，并初始化发送数据包大小payload变量，定义传输信道，并调用config()来配置Arduino：

*Mirf.spi = &MirfHardwareSpi;*

*Mirf.init();*

*Mirf.setRADDR((byte \*)"THIS IS ALEX");*

*Mirf.payload = sizeof(value);*

*Mirf.channel = 90;*

*Mirf.config();*

在发送端loop()函数中，项目使用value表示发送的数据，在使用setTADDR定义发送地址后，通过调用Mirf中的isSending()方法来检测发送状态，使用delay函数来等待发送完成。

接收端Arduino的配置和setup()的实现与发送端非常类似，除了定义和发送端相同的payload大小和通信信道，调用Mirf的config()函数完成基本配置，接收端还需要在setup()时开启串口通信接口并设置好通信的波特率，在这里本项目选用9600的波特率；除此之外，还需要设定好接收端的地址和发送端的目的地址相同。

在接收端的loop()函数中，通过监控Mirf提供的dataReady()的值，一旦为真，表示收到数据且准备好，随即调用getData读取数据。从而完成了从车载安全机器人到中转模块的数据通信。6.3节将介绍如何将中转模块收到的数据通过蓝牙发送给安卓手机。

## 6.3 蓝牙信号的收发

### 6.3.1 Arduino和蓝牙4.0模块的连接方式

下表展示了中转模块（Arduino）和通信模块（HC-06）的连接关系：

表6-2中转通信模块连接方式

|  |  |
| --- | --- |
| Arduino | HC-06（蓝牙4.0） |
| 5V | VCC |
| GND | GND |
| 10 | TXD |
| 11 | RXD |

接入蓝牙模块后整个中转模块如图6-2 所示：

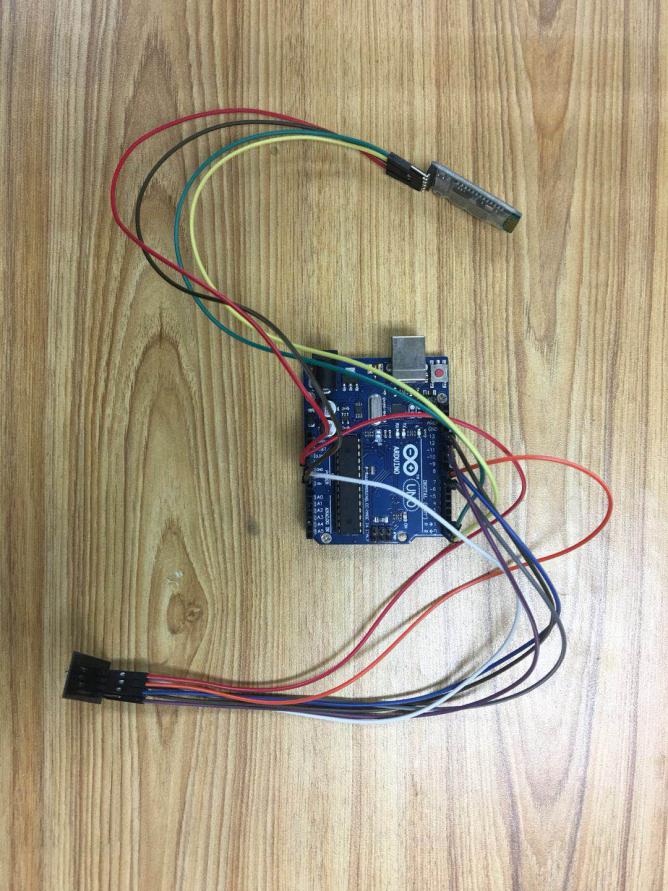


图6-2 连接蓝牙模块

### 6.3.2 蓝牙信号发送端设计与实现

### 6.3.2.1 进入蓝牙模块AT模式

如果要完成蓝牙模块和Arduino单片机的通信，并正确显示收发的消息，必须保证蓝牙模块的收发波特率、代码中初始化的端口波特率以及串口监视器设定的波特率相同。实验的过程中，一开始拿到的蓝牙模块，其配置信息是未知的，因此有必要给出想要的设置，这就需要进入蓝牙模块的AT模式。

将蓝牙模块按照vcc相连，gnd相连，tx与rx互连的方式与串口模块连接，长按蓝牙模块上的复位键，通过USB接入电脑，蓝牙模块的指示灯按照一秒的间隔亮起，表明模块进入AT模式，通过串口助手给蓝牙模块发送设置波特率的at指令：

AT+UART=9600，0，0

当收到回复“OK”，表明成功将蓝牙波特率设置为9600。

### 6.2.2.2 蓝牙端代码设计与实现

蓝牙和安卓手机的通信过程可以描述为：安卓端通过搜索蓝牙设备，发出连接请求，位于Arduino上的蓝牙模块收到连接请求并确认后二者建立起连接。之后蓝牙模块将来自2.4G模块的数据逐一发送给安卓手机，安卓前端再将数据显示出来。

按照策略，蓝牙发送端要转发从2.4G模块收到的数据，由于其转发时设定的波特率和2.4G模块收发时候设置的完全一致，因此不必改变Arduino程序setup()部分的代码。在6.2.2的代码基础上增加转发操作即可。

### 6.3.3 安卓客户端蓝牙模块的实现

由于安卓蓝牙的通信依托的是各类广播的发送和接收，因此在蓝牙模块开启之后，必须注册使用蓝牙的一系列广播。之后编写和各个阶段的广播对应的处理方法。需要注意的是，不同的蓝牙服务对应着不同的UUID（通用唯一识别码），本项目需要实现的是蓝牙的串口服务，因此对应的UUID是确定的： '{00001101-0000-1000-8000-00805F9B34FB}'。接下来本论文将对注册广播、蓝牙配对、数据收发进行介绍：

### 6.3.3.1 广播注册：

整个蓝牙通信过程要完成蓝牙开关、蓝牙匹配、开始和结束蓝牙搜索、蓝牙匹配状态发生变化等操作，其对应的广播为ACTION\_STATE\_CHANGE, ACTION\_FOUND, ACTION\_DISCOVERY\_STARTED, ACTION\_BOND\_STATE\_CHANGED，广播的注册应当在onCreate()函数完成蓝牙模块的可用性检查之后执行。下述的过程可以完成本工程蓝牙广播的过滤以及监听：

新建一个过滤对象*IntentFilter*并给其添加蓝牙监听广播：

*intentFilter.addAction(BluetoothDevice.ACTION\_FOUND);*

添加完所有待监听广播后，要使用相应的注册方法来注册广播接收器。在广播接收模块，对收到的不同广播信号进行分发，执行对应的操作。

### 6.3.3.2 蓝牙配对

用户在前端点击查找设备之后，安卓一旦收到ACTION\_FOUND广播，则表明搜索到新的蓝牙信号，安卓将会把蓝牙的信息显示在前端等待用户进行点击配对。用户点击对应的蓝牙模块后，安卓端将关掉蓝牙搜索，并调用连接设备的getBondState()函数监控配对状态，当状态为未配对则调用安卓蓝牙库函数中的createBond方法，调用invoke()函数建立起蓝牙连接，一旦状态为蓝牙配对成功，则开启数据通信的线程。在新的线程中，安卓端将通过之前定义好的实现蓝牙串口服务的UUID创建一个socket连接，等待Arduino上的蓝牙模块接受请求，连接正式建立后，将开启接受数据的新线程，用于接收来自Arduino的数据并展示在前端。

### 6.3.3.3 数据收发

按照需求分析，本版本的车载安全机器人只完成机器人的位置数据和安全状态数据在前端的显示，因此只需完成数据的接收和显示即可。

使用输入流InputStream，调用已建立的蓝牙socket的getInputStream()即可读到每次发出的数据，按照最开始设定的数据格式，安卓端将读到的数据解析为Json，用于前端位置信息和安全状态信息的显示。下图展示了安卓端获取到的来自Arduino模块中转的数据：

## 6.4 无线数据传输策略评估

在设计无线通信策略时，如果不考虑重写传输协议的复杂性和可行性，本通信策略依然是首选策略，可以从以下几个方面评估考虑：

首先，如果采取智能车直接通过2.4GHz无线网络和手机通信的方式，势必会增大手机的功耗；同时考虑到手机需要腾出WiFi来与智能车维持通信，但此“WiFi”并不能使用户接入互联网，WiFi处于使用状态时安卓将禁用无线蜂窝网络，从而导致用户无法接入互联网。相较之下，如果使用一个连接在汽车上的Arduino做中继，Arduino发送蓝牙数据给手机，一方面中继站可以得到汽车稳定的供电不会消耗手机的电量，另一方面又解决了用户同时使用互联网的需求。

其次，使用一块Arduino连接在车上，在完成整个数据通信中继功能的同时，将整个汽车接入了一个物联网，整个车载安全机器人系统的拓展性也可以得到极大的提升。

# 第七章 系统测试

本章将对照第二章的系统设计，通过实际运行整个工程，开启车载安全机器人使其自动巡航，将测距数据发送会安卓手机。通过检验系统设计中重点关注的巡航模块、测距模块和通信模块这三大模块的工作情况，来验证整个系统工作的正确性和可行性。

## 7.1 测试方法与环境

为验证整个工程是否能满足要求。本项目将装配好的整个车载安全机器人放置在标有道路线的马路上。使用5V/2A电源给树莓派供电，使用12V航模电池给车载安全机器人控制模块供电。测试人员手持安装好且打开了数据接收App的安卓手机，和车载安全机器人站在同一位置

## 7.2 机器人巡航过程测试

上电后，等待15s左右树莓派启动自动运行程序。智能车平稳前进，行进过程中当航线无法与道路线平行，车辆会自动进行微调。当智能车运行规定的时间，到达既定的位置，自动停止。测试结果表明该机器人可以完成自动巡航到达既定位置。

## 7.3 机器人测距和通信过程测试

车载机器人上电运转后，安卓端工作人员连接蓝牙，提示连接成功。

从车载Arduino模块读取到机器人通过2.4GHz通信模块实时发送来的数据。

当小车到达既定位置开始测距，同时，在距离小车百米距离外的工作人员可以从安卓手机上读出当前小车安全状态信息和小车的位置坐标。从安卓客户端的读取结果来看，数据通信过程正常。

当有告诉物体快速靠近小车，可以从客户端数据中读取到danger状态。由此可见该机器人可以通过测距模块正确判断出安全状态，测试通过。

小车读取到的数据情况如图7-1所示：



图7-1 Android端数据接收与显示

## 7.4 本章小结

本章通过对整个工程进行实地测试，通过让小车通电后自动巡航，改变不同的状态，读取数据收发，验证了本次毕业设计完成内容的正确性和可行性。

# 第八章 结束语

## 8.1 论文工作总结

汽车时代高速发展带来了安全隐患，高速公路上发生故障和事故后如何快速高效安全地处理和预警十分重要。本文首先介绍了基于机器视觉的车载安全机器人自动巡航的实现过程，通过机器视觉获取图像，分析图像，抓取图像的特征转化成控制信号，控制智能机器人的自动巡航。同时介绍了一种基于机器视觉的测距方法的实现，并将测距模块安装在了车载安全机器人上面；同时为其开发了安卓客户端用于接收其位置和测距信息用于观测和报警；最后研究并编码实现了2.4GHz通信过程的实现，将整个机器人的通信过程使用2.4GHz协议，完成数据在百米范围的正确收发。

本文取得的主要成果有：

1、将智能机器人巡航过程在树莓派上实现。

2、实现了基于机器视觉的物体测距测速。

3、实现了2.4GHz的通信过程，完成了小车和百米外的安卓手机的无线通信。

本人的主要工作为：

1、自学了智能车的相关知识，自学了树莓派的开发和使用。

2、学习了OpenCV的相关知识，实现了基于机器视觉的智能车控制。

3、自学了安卓、stm32单片机的开发。

4、设计并实现了整个通信过程，使用高效的方式完成了智能车和安卓手机的通信。

## 8.2 问题和展望

本课题将本科阶段自学的计算机视觉的知识学以致用，通过动手搭建测距装置、实验对比与数据分析得出了简洁准确的测距结果；通过自学安卓App的编写，使用安卓手机完成了无线网络通信。总结看来，本测距方式简单可行，受天气等环境因素的影响较小，但有一定的弊端，本项目的测距模块存在可见的激光点，在高速公路上投入使用会使得过往的司机产生困惑，如果未能控制好激光灯照射到了司机的眼睛甚至还会造成比较严重的后果。故将来若真正投入高速公路使用，需要考虑不需要激光灯的算法。

# 参考文献

[1] 张大伟，贺锦鹏，孙立志等. 道路交通事故类型与诱因分析 [J]. 汽车工程师, 2015,01:1

[2] 孙强. 高速公路安全设施设计优化方法研究 [D]. 长安大学, 2011

[3] 郑南宁．计算机视觉与模式识别 [M]．国防工业出版社, 1998

[4] 陈丹．计算机视觉技术的发展及应用 [J]．电脑知识与技术．2008, 12: 2449-2450+2

[5] 尹艺杰. 基于图像识别的车辆距离检测算法研究 [D]. 2012 : 5-8

[6] 邱茂林，马颂德，李毅．计算机视觉中摄像机定标综述 [J]．自动化学报，2000, 01: 13

[7] 成剑. 基于机器视觉的道路线识别算法研究 [D]. 南京理工大学, 2016

[8] 韦唯. 基于单目视觉的车道线检测方法 [C]. 吉林大学, 2015

[9] 韩立成，章回，王永庆. 一种基于2.4G无线通信的智能照明控制系统 [J].中国照明电器 2016, 10 37-39

[10] 周婷婷，尚浩. 基于2.4G的智能家居控制系统设计 [J]. 单片机与嵌入式系统应用2012, 10: 67-69+78

# 致 谢

时光荏苒，四年的大学时光一闪而过。回顾这四年的大学时光，可谓丰富多彩，有轻松欢乐，也有挫折难捱，着实令人难忘。这四年遇到的所有朋友，所有老师，你们给了我太多帮助与鼓励，帮助我进步成熟，感谢一路有你！

在这里，我要特别感谢戴志涛老师，非常荣幸大二时在叶培大学院选择您作为我的导师，您严谨认真的工作态度深深鼓舞了我，在学习研究方面给予了我许许多多的帮助，让我对智能硬件产生了极大的兴趣。本次毕业设计，您更是给了我全力细心的指导。其次，感谢实验室的刘畅学长、宋子恒学长和吕梓宁学长，感谢你们在我毕设进行过程中对我的莫大帮助，没有你们，我很难在这有限的时间里完成这项工程。感谢母校为我提供的良好的学习科研环境，感谢本科四年每一位教过我的老师，毕业后，无论在哪里，我都会谨记厚德、博学、敬业、乐群的校训，我永远都是北邮人！

最后，感谢我的家人，感谢你们对我始终如一的支持和关怀，在今后的日子中，我将更加努力学习工作，努力生活。