

智能车项目设计实验报告

**题 目 基于树莓派的公路车流量统计系统开发**

**电信 学院 自动化 系**

**学生姓名 刘渝**

**班级 自动化钱91 学号 2191411802**

**指导教师 刘小勇**

2023年3月

**实验题目：基于树莓派的公路车流量统计系统开发**

**学生姓名：刘渝**

**指导教师：刘小勇**

摘 要

四旋翼飞行器结构较为简单，运动灵活，收到国内外研究者广泛关注。该系统具有六个自由度，四个控制输入，是非线性强耦合系统，因此研究出一个高性能的四旋翼控制系统非常复杂，而这也是研究四旋翼飞行器的关键。

我们建立了四旋翼无人机的动力学模型，采用STM32F10RCT6单片机作为我们的主控单元，通信模块采用SPI协议的NRF24L01实现与我们的遥控器之间的数据传输，姿态采集模块在MPU 9250的基础上设计了软件滤波模块，利用四元数描述四旋翼飞行器空间旋转，利用姿态矩阵求解出欧拉角，采用双闭环的PID控制四旋翼飞行器姿态。在基本的飞行稳定实现之后，我们还设计了了利用树莓派处理摄像头获取的二维码信息并且与STM32实时串口通信从而控制四旋翼飞行器的运行系统的方案，从而实现视觉识别的功能，更具智能化。

**关 键 词**：STM32、四旋翼飞行器、NRF通信、欧拉角、双闭环PID、二维码识别

目 录

[1 实验简介及原理介绍 4](#_Toc105106781)

[1.1实验简介 4](#_Toc105106782)

[1.1.1实验背景 4](#_Toc105106783)

[1.1.2实验目的 4](#_Toc105106784)

[1.1.3实验内容 4](#_Toc105106785)

[1.2实验原理 4](#_Toc105106786)

[1.2.1四旋翼无人机发展 简介 4](#_Toc105106787)

[1.2.2四旋翼无人机飞行器结构以及原理 5](#_Toc105106788)

[1.2.3基于STM32的四旋翼整体架构 7](#_Toc105106789)

[2 硬件及其端口 9](#_Toc105106790)

[2.1 STM32F103RCT6 9](#_Toc105106791)

[2.2 NRF24L01通信模块 10](#_Toc105106792)

[2.3 MPU9250陀螺仪 11](#_Toc105106793)

[2.4无刷电调 11](#_Toc105106794)

[2.5电池 12](#_Toc105106795)

[2.6无刷电机 13](#_Toc105106796)

[2.7摄像头 14](#_Toc105106797)

[3 软件程序 15](#_Toc105106798)

[3.1主函数 15](#_Toc105106799)

[3.2控制算法 15](#_Toc105106800)

[3.2.1 PID可能的问题 15](#_Toc105106801)

[3.2.2串级PID的思路 16](#_Toc105106802)

[3.3其他处理 16](#_Toc105106803)

[3.3.1遥控器数据解算 16](#_Toc105106804)

[3.3.2 MPU9250的处理 17](#_Toc105106805)

[3.3.3 NRF24l01模块通信 17](#_Toc105106806)

[3.3.4安全解锁 18](#_Toc105106807)

[3.3.5立即停止 18](#_Toc105106808)

[3.4视觉识别 18](#_Toc105106809)

[4 调试说明 21](#_Toc105106810)

[4.1调试过程 21](#_Toc105106811)

[4.2调试中的问题 21](#_Toc105106812)

[4.3.1忘记找老师借用调试架 21](#_Toc105106813)

[4.3.2 STM32烧写困难 21](#_Toc105106814)

[4.3.3无刷电机动力过于强劲 21](#_Toc105106815)

[4.3.4硬件支持有时会出问题 22](#_Toc105106816)

[5 心得体会 23](#_Toc105106817)

[6 附 录 24](#_Toc105106818)

在每一章的末尾插入下一章的MathType的章标记（打印前将其字体颜色变为白色，在打印预览中看不见即可）：111Equation Chapter 1 Section 1

# 项目简介

## 1.1项目背景

随着道路上车辆数量不断增加，人工统计公路段汽车流量难度增大，成本渐高。而公路车流量数据应用方面十分广泛。如规划道路交通，建设智慧城市等方面都可以以公路车流量数据作为统计分析的基础。

因此，探索出有效的、成本较低的公路车流量检测技术，对城市的生产和服务效率的提升有很大价值。

交通规划部门对于公路车流量检测，常用的技术包括空气管道检测、磁感应检测（地感线圈传感器、磁阻传感器）、波频检测等[1]。这些检测技术或需要安装繁琐设备，破坏已建路面，或受场地因素等外在条件影响较大，稳定性不高。在自动化、数字化的新时代，传统检测方法很难满足要求。因此，在公路车流量统计方面，由于视频检测技术具有易于数字化、实现简通、普适性强等诸多优点，使用低成本的电

子设备拍摄图像，检测车监控和估计交通流量的方式是大势所趋。

## 1.2相关工作

目前，我国这方面研究工作尚不能兼顾设备的低成本与算法的高精度、高可靠性。陈佳倩等[2]在实验配置环境为英伟达 RTX 2060 SUPER以及 Win10 系统的情况下，采用YOLOv3 和 DeepSort 框架对于车辆检测有较高的准确率；陈璐等[3]在树莓派和Movidius神经元计算棒的运算环境下，使用MobileNet+SSD 的算法也对各种环境下的车辆有较好的检测功能；白涛等[4]通过树莓派捕获视频流，上传至阿里云平台进行车辆识别检测模型的计算。以上的工作均依靠着高成本、特殊的设备或者云计算平台来进行车辆检测与识别，很难在成本受限的情况下满足普遍场合的需求。

## 1.3本文工作

本文通过在树莓派系统上应用背景差分法中的混合高斯模型算法，能够较好地识别出前景检测目标（即汽车），并针对雨天、夜晚等环境作出了算法的修正改进.

222Equation Chapter (Next) Section 1

# 实验环境

## 2.1 硬件设备

本文采用 CPU 型号为 BCM2835 的树莓派 3B+。这款树莓派体积小、功能全面且搭载 Linux 操作系统，具有强大数据处理能力。树莓派外接视频采集终端、电源等接口，视频采集终端选用 CSI 接口的摄像头模块，相比 USB 摄像头， CSI 接口配置的摄像头价格低廉，符合本文需求；同时，500万像素确保能够精准识别获取到的公路车信息，在正常的情况与雨天、夜晚等特殊情况下都能够实时捕获视频流。

## 2.2 软件系统

本文设计的算法程序主要在树莓派对应的 Linux 操作系统环境下进行调试；具体方法是采用VMware16+ubuntu18.4运行图形化Linux界面。编程语言采用Python。

在实地应用中，会采用并行线程处理方式对视频流进行处理。并行线程处理方式会保证系统的稳健性，并且能够充分利用系统资源，高效地处理视频流信息。

## 2.3 数据集

在实验调试过程中，采用信路威视频数据集，网址为 http:// gofile.me/6IVQe/WFWoIsz Hz.

333Equation Chapter (Next) Section 1

# 基本算法与整体思路

软件部分主要分为控制算法和视觉识别两部分来进行介绍，控制算法依托STM32进行，视觉识别则依托树莓派使用opencv库进行。

## 3.1高斯混合背景建模

高斯混合背景建模基于以下的设想：

1、背景在一定时间内应当是固定不变的，或者随时间缓慢变化，即背景变化比较小

2、摄像头获取的图像通过像素值进行表示

因此对于各像素值可以通过高斯分布来描述它，当像素值按一定概率符合分布时，认为是背景，否则就认为是前景

通过维护多个高斯分布的均值、方差和置信度，置信度之和为一用于表示对背景的信心，形成对于背景的描述

每次获取新帧时，将对应位置像素与维护的高斯分布对比，如果符合某个分布就计入该分布的置信度，如果置信度高于一定值就认为是背景，否则为前景

高斯分布的更新方法如下：

1、如果当前像素与维护的分布有符合的，就更新符合的分布参数，提高其置信度

2、如果当前像素与维护的分布均不符合，就去掉置信度最低的分布，并根据当前像素初始化一个新的分布

## 3.2帧差法

初始背景建模

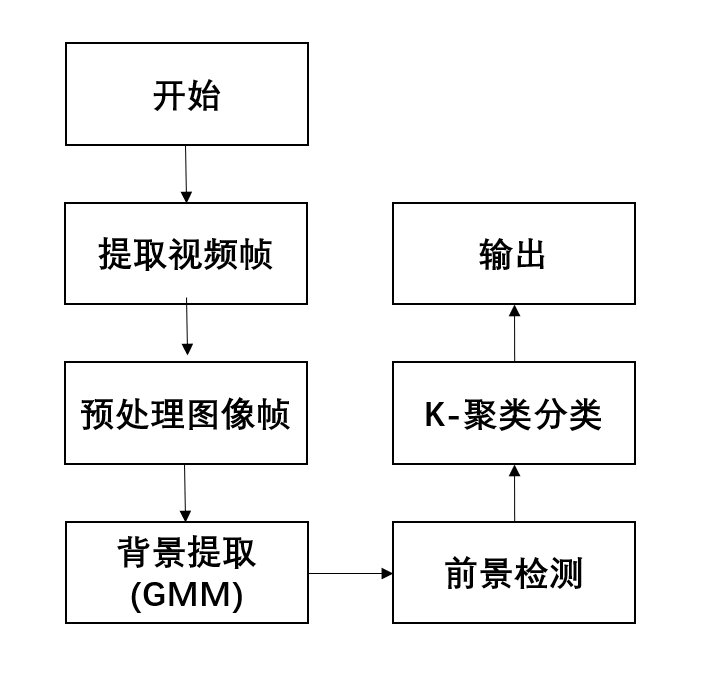
首先通过读入一定数量的帧对场景建立一个初始背景

分块帧差粗检测

分块计算两帧差值，统计一整个块的差值绝对值之和，如果整块差值大于指定阈值，则认为发生了运动，否则认为没有运动

## 3.3整体系统

为了方便区分工作，使编程思路更清晰，对整个系统的工作流程进行总结



系统工作流程总结

# 程序实现要点及实现结果

## 4.1混合高斯背景建模

尽管Opencv提供混合高斯背景建模的函数，但是我根据其原理自己写了一个C++的实现。由于实现占用过多运行内存，运行时常出现栈溢出的情况，因此已经取消掉这种实现，放弃了该版代码。

这里简单说明该版实现的要点：

维护一个N\*M\*C的数组，N为图像的高度，M为图像的宽度，C为背景建模的数量，由于维护的是高斯概率函数，需要一个均值和一个标准差，并根据原理维护一个置信度

然后根据每帧读入的图像对数组进行更新：

如果像素值不属于维护的任何一个分布，将其加入或替换置信度最低的分布

## 4.2帧差法

利用一个数组记录各块的差值之和

## 4.3记录车辆

维护一个记录包围车辆的矩形的数组，在每一帧进行识别和更新，如果当前帧中的矩形与维护数组中的矩形中心点距离小于指定值且面积差值小于指定值，则认为当前帧矩形与记录中的为同一辆车，则更新数组；否则认为当前帧识别的矩形为新出现的车，在维护的数组中新加入一个矩形进行记录；一定帧数未得到更新的矩形认为已经行驶出了摄像头监测区域，将其有效标志位置0表示出范围

## 4.4上下行、大小、颜色

### 根据维护的记录车辆的数组进行判断

上下行：

设置一个数组记录每辆的初始中心点（进入摄像头画面），若驶出的中心点y方向的值小于起始位置，则记录为下行，否则为上行

大小：

由于距离摄像头的距离会影响车辆所占像素的面积，因此设置一个数组记录每辆的初始面积，如果相对摄像头上行则利用初始面积判断车辆大小，反之则利用车辆的结束面积

颜色：

在记录过程中对车辆进行截图，然后根据HSV色域来区分并统计各颜色像素的数量，选择像素数量最多的颜色作为车辆的颜色，如果仅区分深色和浅色则程序可以更稳健一些

# 问题说明及总结

## 5.1问题说明

白天情况：

1、车辆密集聚集在一起无法区分

提取前景的方法没有特征的标记功能，仅能识别图像中的某些像素是否发生变化，因此聚集在一起的车辆将会被提取为一辆车，但一定的膨胀操作是必要的，否则会影响到车辆识别结果的闭合，使识别效果不佳

通过记录车辆的消失设想可以解决这个问题

晚上情况：

1、晚上车辆会打开车灯，车灯会在车辆前方投出大片的亮块，亮块会被显著地识别为正在运动的前景，从而无法正确识别；而同时车辆本身比较暗，在识别上有一定缺失

因此我们决定使用车灯来进行识别，车灯间距及大小用于确定车的大小

2、然而采取车灯进行识别存在以下问题：

车灯打出的光会在车的表面发生反射，由于车身的金属材质这种反射很强，而且通常还是对称的，即车身上左右各有一块有一定大小的亮斑

这使得反射的亮斑具有了车灯的一切性质：对称、大小、间距

针对这些特点加入的规则通常很容易在另外的视频失效

## 5.2总结

通过传统图像识别方法可以搭建出识别车辆的系统，但是由于现实情况的极端复杂，系统的普适性还有待进一步加强。更有可能的应用场景是在某些特定的场所，出入的车辆都比较固定，通过白名单方法设置模板进行匹配，才能鲁棒地识别。不过鉴于大多数普通的非定制的车辆造型都比较相似，通过设定模板匹配应当也有助于提高识别效果。

# 参考文献

飞控代码：

运动控制：

#include "stm32f10x.h"

#include "motion\_control.h"