

浙江大学

探究性实验项目总结报告



课程名称	<u>智能控制系统设计与实践</u>
实验项目	<u>基于摄像头和电磁传感器的智能循迹控制系统设计</u>
姓 名	<u>刘科明</u>
学 号	<u>3170101195</u>
专 业	<u>电子信息工程</u>
时 间	<u>2020 年 08 月 30 日</u>

一、 实验目的和要求

1. 了解车模构成，掌握车模组装方法；
2. 掌握 MCU 的驱动和软件开发；
3. 掌握常用传感器的驱动方式，如摄像头、编码器、陀螺仪等；
4. 掌握 PCB 的绘制和电路板的焊接、调试；
5. 掌握图像校正、提取黑线、赛道元素识别的方法；
6. 掌握模糊控制、PID 控制的原理和参数调试方法。

二、 实验内容

2.1 控制器选型

方案一：采用 LPC54606 芯片。该芯片采用 Cortex-M4 内核，主频 180MHz，SRAM 为 200K，Flash 为 512K，对于小车图像处理算法性能不足。

方案二：采用 i.MX RT1064 芯片。该芯片是 i.MX RT 系列，由 NXP 半导体公司推出，是一款跨界处理器芯片。该系列下又包括 i.MX RT1020、i.MX RT1050 及 i.MX RT1060 等子系列芯片。该芯片采用了微控制器的内核 Cortex-M7，对中断响应更快，具有良好的实时性，从而具有应用处理器的高性能及丰富的功能，又具备传统微控制器的易用、实时及低功耗的特性。其主频高达 600M，且板载 4M 大容量片上 Flash 和 512KB 的 RAM，满足程序开发的需求。^[1]

因此最终选择方案二，采用逐飞 RT1064 核心板。

2.2 摄像头选型

方案一：采用 OV7725 二值化摄像头。虽然其数据量小，易于单片机处理，但由于竞赛中摄像头的高度要求 10cm，该摄像头已经不再满足新的赛道条件的需求。

方案二：采用 MT9V032 灰度摄像头。其拥有 140° 广角视域，帧率高。因此图像更新速度快，采集信息量大，满足比赛需求。

因此结合上述分析，最终选择方案二，摄像头采用总钻风 MT9V032，帧率 150FPS。

在实际应用时，摄像头帧率设置过高会导致图像变暗，且过高的帧率会超出程序控制周期，即图像更新的信息在控制时可能不被采用。此时可通过配置摄像头增益等参数对图像进行优化，根据实验效果选择较优的配置。

2.3 屏幕选型

方案一：采用 OLED 显示屏。传输数据量小，实时性高。但分辨率较低，尽可显示二值化图像。

方案二：采用 IPS 显示屏。分辨率更高，能显示摄像头原图数据，但是数据是 16 位灰度数据，传输数据量非常大。在显示图像和变量时，由于软件 SPI 延时，图像处理程序仅为 20FPS，不满足智能车高速运行时的帧率需求，所以 IPS 屏仅能在静止调试时使用。

因此最终选择方案二，采用 IPS 显示屏。

2.4 智能车系统整体方案

1. 采用 MT9V032 摄像头，以 150fps 的帧率拍摄赛道图像。图像数据通过 DMA 自动传输（场中断传输和行中断传输）给 RT1064，然后运行程序对图像进行反光点去除、边缘检测+二值化预处理、校正、分割并提取黑色边线，并经过路径规划得到小车的预期路径，并判断赛道类型。^[2]
2. 由规划好的路径，选择图像合适位置的参考行，计算出偏差，用模糊 PD 控制舵机转向；当进入坡道时，由于拍摄到赛道外图像，故切出摄像头状态，使小车保持直行，离开坡道恢复。
3. 通过编码器检测车速，RT1064 输入捕捉功能进行脉冲计数，2ms 的 pit 中断测量。通过脉冲计数及时间，可计算得到小车的实时速度。通过路径偏差计算出预期速度，并利用 PI 控制以及 bang-bang 控制算法调节电机输出，从而实现小车速度控制。
4. 采用陀螺仪进行坡道辅助判断。记小车车头朝向为 x 轴正方向，通过 I2C 协议向陀螺仪 ICM20602 的相关寄存器中实时读取当前 y 方向的角速度值，以获得小车的翻转角速度，从而判断当前是否正在经过坡道或者颠簸。由于小车撞到路肩时同样可能出现翻转角，因此加入防误判算法。
5. 向 UART 转 Wi-Fi 模块以 150000 的波特率发送串口数据。模块将数据传输回上位机，实现实时监控，从而获得小车运行过程中的各项观察指标，便于后续参数调节。
6. 采用拨码开关、按键以及 IPS 显示屏辅助进行控制策略选择和参数设置。

系统框图如下：

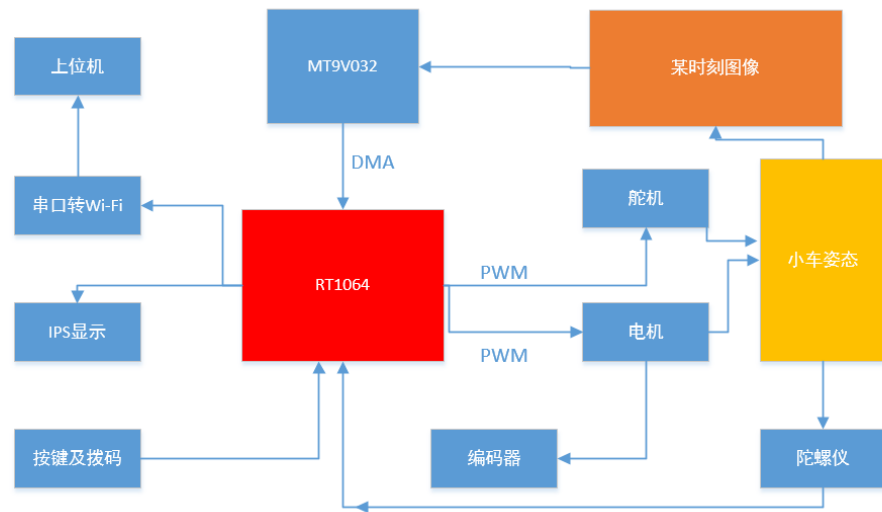


图 1 系统框图

三、 技术要点

第一节 电路设计

电路设计采用了主控板和驱动板分开的方式，这样一方面减少电机驱动部分电路对主控的干扰，另一方面缓解了布线压力，所有电路板都可以采用双面板。之后我们对电路板的设计进行了优化，将主板以及电机驱动板均做成了 4 层 PCB，驱动板的面积缩减为不到原来的 1/2，主板的面积缩减为原来的 3/4。更小的面积有利于电路板的安装。

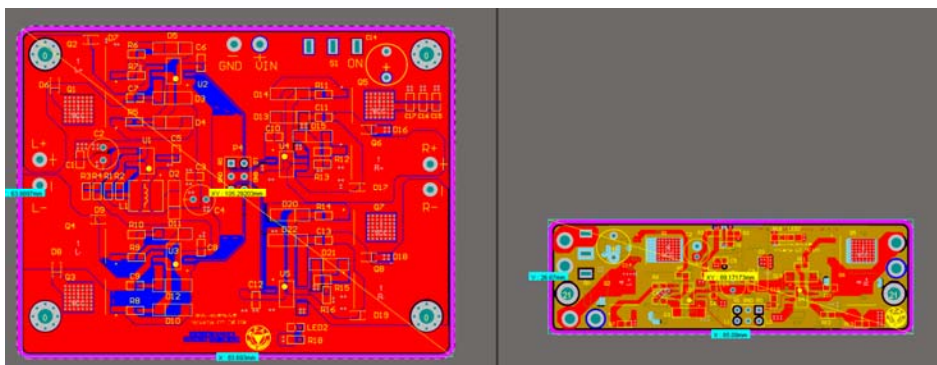


图 2 原驱动板（左）与新驱动板（右）

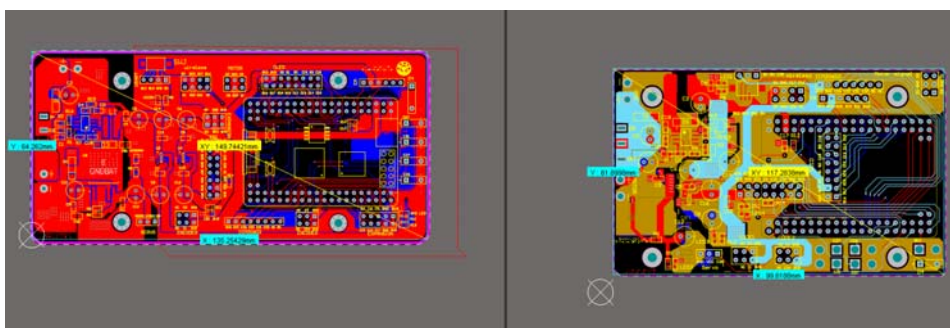


图 3 原主板（左）与新主板（右）

由于 RT1064 芯片外围电路较为复杂，主频较高，对电源质量、时钟信号走线都有很高要求。因此，相比绘制完整主板，直接采用官方核心板无疑是一个更好的方案。

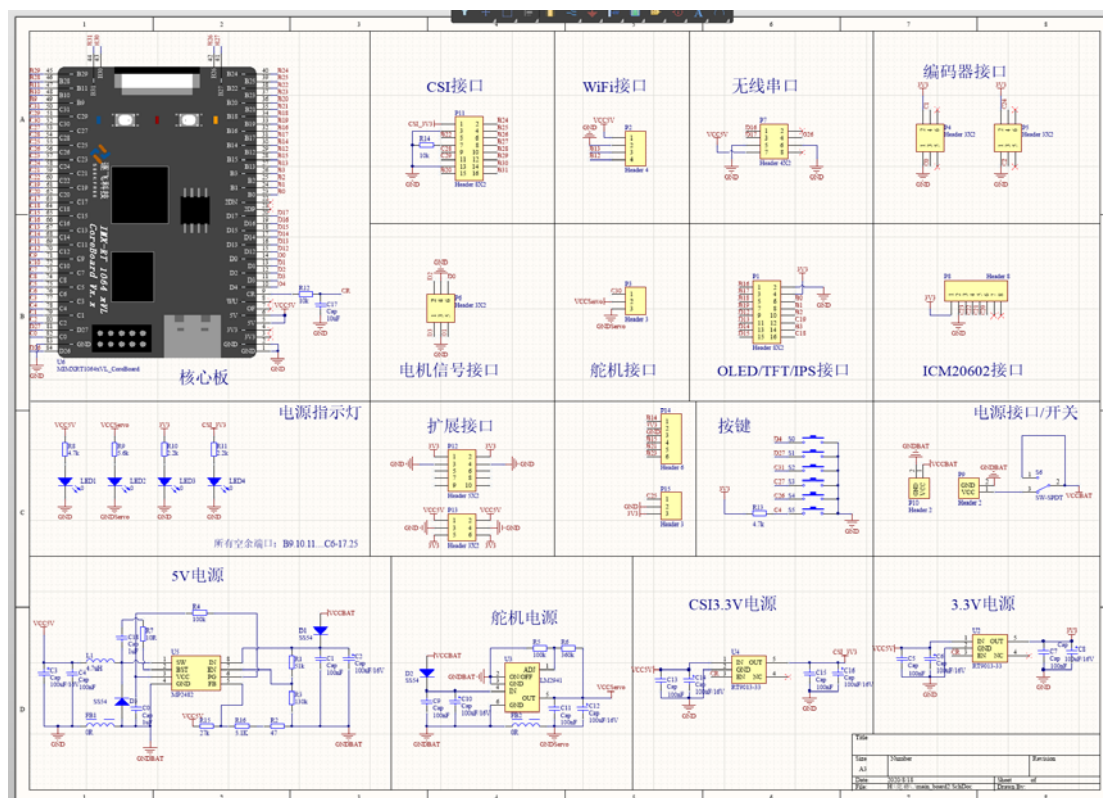


图 4 主板原理图

- **主板：**首先采用 MP2482 进行电源电压到 5V 的稳压，相比以往采用的 LDO，该芯片为 buck-boost，在电源电压被电机堵转拉低时，还有较好的稳压能力和输出能力，该芯片相较于 TPS63070 最大的优势在于封装为 SOC-8,非常易于焊接，而 63070 的封装为 VQFN，即无引脚封装，焊接起来存在一定困难。其输入电压最低为 4.5V，输入电压范围比 TPS63070

小，但已经可以基础四轮组对于电源的要求。

- **3.3V 稳压：**采用了两路 RT9013-33LDO 线性稳压器，一路供给摄像头，一路供给 IPS 屏幕等传感器。该稳压器电路十分简单，且易于焊接调试。
- **舵机电源：**采用了 LM2941 提供 6V 的电压，该方案能够提供的最大输出电流为 1A，LM2941 的限流能力能够很好的保护舵机。

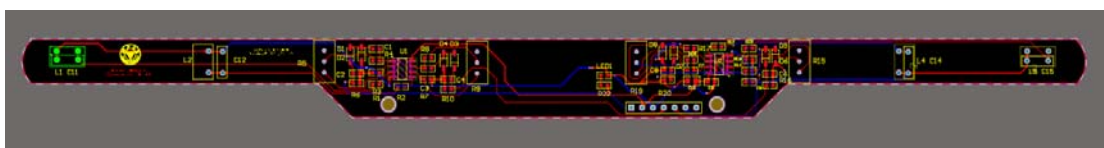


图 5 电磁杆

- **电磁运放电路：**选择 OPA 系列的运放，为节省空间将运放电路放置在电磁杆上，此设计可以节省主板空间。
- **驱动电路：**沿用最常用的方案，2104（驱动）+7843（MOS），性能稳定，驱动需要一个 12V，因此采用的是 34063，该升压电路的反馈电阻（ 0.22Ω ）一定要保证阻值正确，且电容不能过大，否则电路工作不正常。另外，要保证通过大电流的通路的铺铜宽度足够。[5]

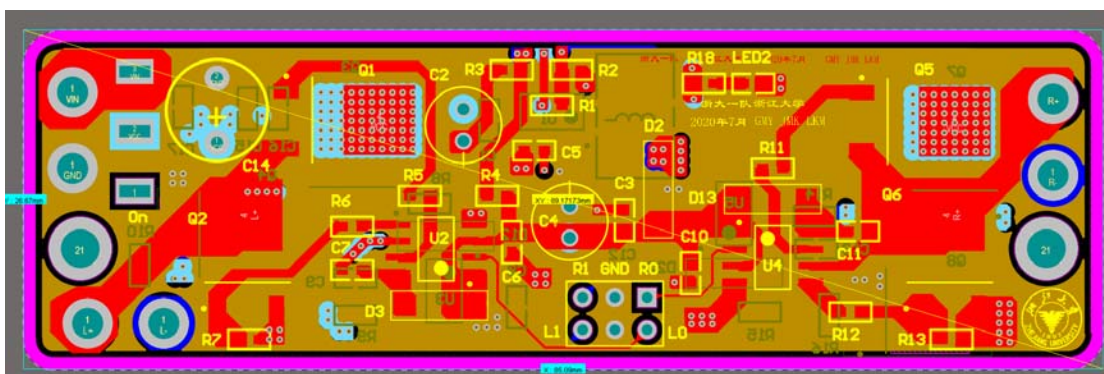


图 6 驱动板

电路设计以稳定、通用、成熟方案为宗旨，因为智能车电路不涉及过多的 PCB 设计知识，只要布局合理，电源处理恰当，即可工作正常，所以尽量不要追求过分个性化的设计，比如异形 PCB，过度密集的布局、过分追求更小等。

第二节 电磁信息获取

2.1 理论模型

由于电磁传感器放置位置受限于摄像头视野范围，仅能达到 3-5cm 左右的前瞻。为矫正传统偏差计算方法带来的误差，改进经典电磁模型以适配传感器状态。

首先一根无限长通电导线的磁感应强度与导线距离的平方成反比：

$$B = \frac{B_0}{r^2} \quad (\text{公式 1})$$

赛道电磁线在小尺度下可看作无限长通电导线。且其中通交流电，所以通过电感的磁感应强度及对应感生电流也是交变的。记电感到电磁线距离为 l ，则有感生电流的大小 $I_{\text{感应}}$ ：

$$I_{\text{感应}} \propto 1/l^2 \quad (\text{公式 2})$$

使用距离平方的反比代替小车测得的电感电压。实际系统中将交变电流做了交-直变换后得到了电感电压，由于变换线性，故测量值仍与距离平方反比成正比例关系。

记电感距离地面电磁线的高度为 h ，左右电感相对距离为 n 。则有两侧电感

测量值 $u_{\text{测量}1}$ 、 $u_{\text{测量}2}$ 对应表达式：

$$\begin{cases} u_{\text{测量}1} \propto 1/(h^2 + a^2) \\ u_{\text{测量}2} \propto 1/(h^2 + b^2) \\ a + b = n \end{cases} \quad (\text{公式 3})$$

以上公式未考虑电感与电磁线非完全垂直的情况，受电感安装位置影响，其更易随车头而左右晃动，使得测量值出现较大波动。此时需要针对角度量执行矫正，减小测量误差。

在引入角度偏差的情况下，上述公式修正为[3]：

$$\begin{cases} u_{\text{测量}1} \propto \frac{1}{h^2 + a^2} * \cos(\theta) \\ u_{\text{测量}2} \propto \frac{1}{h^2 + b^2} * \cos(\theta) \\ a + b = n \end{cases} \quad (\text{公式 4})$$

其中 θ 代表电磁线偏离小车电磁感法向量的角度。

假设在测量点引入另一个电感，该电感垂直于原电感，则同理有电感测量值公式：

$$\begin{cases} u_{\text{测量}1}' \propto \frac{1}{h^2 + a^2} * \sin(\theta) \\ u_{\text{测量}2}' \propto \frac{1}{h^2 + b^2} * \sin(\theta) \\ a + b = n \end{cases} \quad (\text{公式 5})$$

因此理论上只需要将两个测量值相除直接可得 θ 的正切值，但是实际系统中，由于并不是理想的无限长直导线，以及测量值较小时存在较大测量误差，因此应用中选择测量值相对大的一组电感相除来计算 θ 角。由此我们可以得到该偏差角。此外，如果假设车辆轨迹是理想的电磁线切线，该角度反馈为弯道曲率，可用于前文所述主动差速的输入。

因为在上述方程组中仅有 a 和 b 是未知量，而求取 $\frac{(a-b)}{2}$ 也就是中线偏差是主要目的，所以可以直接得到这个偏差正比于一个量：

$$\frac{(a-b)}{2} \propto \left(\frac{1}{u_{\text{测量}1}} - \frac{1}{u_{\text{测量}2}} \right) * \frac{\cos(\theta)}{n} \quad (\text{公式 6})$$

再结合前述 θ 表达式：

$$\cot(\theta) = \begin{cases} \frac{u_{\text{测量}1}}{u_{\text{测量}1}'} & u_{\text{测量}1}^2 + u_{\text{测量}1}'^2 > u_{\text{测量}2}^2 + u_{\text{测量}2}'^2 \\ \frac{u_{\text{测量}2}}{u_{\text{测量}2}'} & u_{\text{测量}1}^2 + u_{\text{测量}1}'^2 < u_{\text{测量}2}^2 + u_{\text{测量}2}'^2 \end{cases} \quad (\text{公式 7})$$

即可求得中线偏差从而给后级舵机 PD 控制使用。

上述模型中依然存在一些误差：首先两个放置很近的电感存在无法抵消的互感现象，对称布局三个电感、使用铝箔电磁屏蔽等方法对解决该问题效果不佳；另一方面由于电感体积较大，两个电感并非在同一点测量场强，造成测量误差。^[3]

2.2 系统辨识

在实际应用中，利用左右电感的单侧或双侧值，可以得到小车中心线距赛道重心线的距离。根据理论推导可知，小车电感的采样值与小车与中心线距离成反向单调趋势，故可以通过分段线性拟合的方式拟合出小车电感值与小车中心线距赛道中心线的距离。

由于两电感安装位置有一定距离，故两个电感不会完全饱和。因此，根据单侧值或双侧值可以很容易拿到小车的位置。

然而，此种方式在赛道前方有弯道、十字、圆环等赛道类型时，该方法

并不适用。但在短坡道上采用控制小车向前行进，电磁导航是足够稳定的。^[4]

2.3 电磁保护

小车在高速行进过程中，遇到撞击、翻车等事故对小车的机械结构及硬件设备都会带来很大的损坏，而赛道外无电磁的故有故有特征可以用来做小车冲出赛道的判断条件，以及时刹停小车。但也要防止小车飞离赛道或车较偏时的误判定。

第三节 其他传感器

陀螺仪在四轮光电组中速度控制方面并不是很重要，但是在上坡识别中，可以采用图像、激光和陀螺仪的方法，其中采用图像较为麻烦，采用激光又存在误判的情况，且在上坡过程中，采用陀螺仪积分的方式配合电磁打角比其他方式更加具有可控性，所以采用陀螺仪是非常有必要的。

我们选用的数字式陀螺仪 ICM20602，它支持 I2C 和 SPI 接口，可采用库中的硬件 SPI 配置和采集方式，数据稳定。事实证明 ICM20602 比常见的 MPU6050 性能稍好。同时，我们还采用了卡尔曼滤波[4]，对裸数据进行了标定，实际数据准确，上坡检测效果好。

陀螺仪判定坡道会存在两个问题，第一个问题是小车压到路肩时会出现误判，故还需加入误判检测，这里不再详细赘述；另一个问题是陀螺仪判定坡道时需要对角度进行判定，梯形坡道和圆弧坡道的特征不同，也需要根据实际情况对参数进行修改。

第三节 其他

此外，还有机械设计、图像信息获取、方向控制、速度控制等其他技术要点，此处不再赘述。

四、 实验成果

- 此次比赛中，获得全国大学生智能汽车竞赛（浙江赛区）第七名的成绩。
- 此次比赛中，共历时 11 个月，起止时间：2019.09.20-2020.08.10
- 此次比赛中，共撰写 RT1064 嵌入式 C 语言代码 2w 行（修改的库函数除外）。最终输出 C 语言代码 1.1w 行（库函数除外）。
- 此次比赛中，共撰写 Matlab 仿真代码 1k 行，Simulink 仿真工程 2 份。
- 此次比赛中，共输出比赛日志 100 天共 1.2k 行 2w 字。
- 此次比赛中，共输出电路板原理图及 PCB 工程文件 8 组。共输出电路板 100 余块，成品电路 20 余块。
- 此次比赛中，共输出 1w 字技术报告 1 份。

五、 实验体会

- 我本次主要参与整体搭建，完成电路设计、电磁信息获取等。在电路设计方面，我更加熟悉了多层板布线方法及技巧，学会如何更合理的走线和铺铜以减小电路板体积、增强电路板稳定性。在硬件设计调试过程中，也对 H 桥驱动，以及其他稳压电路有了更多学习和理解。在电磁信息处理上，我使用 Matlab 对参数进行标定，加快了参数调整的过程。通过此次实践，我对硬件设计和电磁信息都有了更深的理解。
- 在实践的过程中，虽然我们各自负责不同的模块，但车辆的整体调试是一

个系统工作，各个控制器、常量参数都会对小车动态产生影响，本次实验深度加强了我们对经典控制理论、对模糊控制等控制方法的实践认识，同时在此基础上对嵌入式软件设计又有了新的理解。

参考文献

- [1]卢守义,万星,卜令坤.基于 CMOS 摄像头的智能车设计[J].自动化应用,2018(04):138-140.
- [2]李永,冯伟峰,李思光,王俊人.基于 MT9V032 摄像头的智能车软件设计[J].火炮科技与市场,2020(01):250.
- [3]陈国定,张晓峰,柳正扬.电磁智能车电感排布方案[J].浙江工业大学学报,2016,44(02):124-128.
- [4] Sui Jinxue. " NXP Cup Intelligent car Design and Example Tutorial [M]. Beijing: Electronics Industry Press. 2018.8.
- [5] C. Kinnaird, "Digital adjustment of DC motor drive circuit parameters," 2016 IEEE Dallas Circuits and Systems Conference, Arlington, TX, pp. 1-4, 2016.

附 A 模型车的主要技术参数说明

改造后的车模总重约 882 克，长约 25 厘米，宽约 19.5 厘米，高约 11.5 厘米（其中摄像头中心高度约 9.5 厘米）。

车模静止状态下电路平均功耗 1.75W，电容总容量 2mF。

车模包含四类传感器：摄像头（1 个）、编码器（2 个）、陀螺仪（1 个）以及电磁传感器（含 4 个电感）。

除车模自带的驱动电机以及舵机外，未使用其它的伺服电机。

赛道信息检测频率与摄像头采样频率相同，约 150Hz；检测精度良好。

附 B 主要代码

1 大津法图像二值化阈值计算

```
1. uint8_t GetOSTU(void)
2. {
3.     int16_t i, j;
4.     uint32_t Amount = 0;
5.     uint32_t PixelBack = 0;
6.     uint32_t PixelIntegralBack = 0;
7.     uint32_t PixelIntegral = 0;
8.     int32_t PixelIntegralFore = 0;
9.     int32_t PixelFore = 0;
10.    double OmegaBack, OmegaFore, MicroBack, MicroFore, SigmaB, Sigma; // 类
    间方差;
11.    int16_t MinValue, MaxValue;
12.    uint8_t Threshold = 0;
13.    uint16 HistoGram[256]; //
14.    int16_t Mincount = 0, Maxcount = 0;
15.
16.    for (j = 0; j < 256; j++)
17.    {
18.        HistoGram[j] = 0; //初始化灰度直方图
19.    }
20.
21.    for (j = START_LINE; j < CAMERA_H; j++)
22.    {
23.        for (i = 0; i < CAMERA_W; i++)
24.        {
25.            HistoGram[Image_Use[j][i]]++; //统计灰度级中每个像素在整幅图像中的
            个数
26.        }
27.    }
```

```
28.
29.     //获取最小灰度的值
30.     for (MinValue = 0; MinValue < 256 && Histogram[MinValue] <= 5; MinValue++)
31.     {
32.         Mincount += Histogram[MinValue];
33.         Histogram[MinValue] = 0;
34.     }
35.     //获取最大灰度的值
36.     for (MaxValue = 255; MaxValue > MinValue && Histogram[MaxValue] <= 15; MaxValue--)
37.     {
38.         Maxcount += Histogram[MaxValue];
39.         Histogram[MaxValue] = 0;
40.     }
41.     //滤除反光点
42.     for (j = START_LINE; j < CAMERA_H; j++)
43.     {
44.         for (i = 1; i < CAMERA_W; i++)
45.         {
46.             if (Image_Use[j][i] > MaxValue - 8)
47.             {
48.                 Image_Use[j][i] = MaxValue - 8;
49.             }
50.         }
51.     }
52.
53.     Histogram[MaxValue] += Maxcount;
54.     Histogram[MinValue] += Mincount;
55.     if (MaxValue == MinValue)
56.     {
57.         return MaxValue; // 图像中只有一个颜色
58.     }
59.
60.     if (MinValue + 1 == MaxValue)
```

```
61.    {
62.        return MinValue; // 图像中只有二个颜色
63.    }
64.
65.    for (j = MinValue; j <= MaxValue; j++)
66.    {
67.        Amount += HistoGram[j]; // 像素总数
68.    }
69.
70.    PixelIntegral = 0;
71.    for (j = MinValue; j <= MaxValue; j++)
72.    {
73.        PixelIntegral += HistoGram[j] * j; //灰度值总数
74.    }
75.
76.    SigmaB = -1;
77.
78.    for (j = MinValue; j < MaxValue; j++)
79.    {
80.        PixelBack = PixelBack + HistoGram[j];
81.        PixelFore = Amount - PixelBack;
82.        OmegaBack = (double)PixelBack / Amount;
83.        OmegaFore = (double)PixelFore / Amount;
84.        PixelIntegralBack += HistoGram[j] * j;
85.        PixelIntegralFore = PixelIntegral - PixelIntegralBack;
86.        MicroBack = (double)PixelIntegralBack / PixelBack;
87.        MicroFore = (double)PixelIntegralFore / PixelFore;
```

```
88.      Sigma = OmegaBack * OmegaFore * (MicroBack - MicroFore) *  
      (MicroBack - MicroFore); //计算类间方差  
89.      if (Sigma > SigmaB)  
          //遍历最大的类间方差 g      //找出最大类间方差以及对应的阈值  
90.          {  
91.              SigmaB = Sigma;  
92.              Threshold = j;  
93.          }  
94.      }  
95.      return Threshold; //返回最佳阈值;  
96. }
```

2 转向控制策略

```

1. void      Turn_Cam_dias(void)
2. {
3.     float   temp;
4.     static  float   car_straight_dias_old;
5.
6.     car_straight_dias    =    M_Slope_fig()    *
    SERVO_DIVIDE_ANGLE_SCALE;
7.
8.     Straight_offset_filter();
9.     car_center_dias    =    car_center();
10.    Center_offset_filter();
11.    if(Road    ==    0    &&    Road0_flag    ==    0    &&
    Road0_flag0_flag    &&    fabs(car_straight_dias    -
    car_straight_dias_old)    <    30    )
12.    {
13.        temp    =    car_straight_dias    +
    PID_CAR_STRAIGHT_CAM.D    *    (car_straight_dias    -
    car_straight_dias_old);
14.    }
15.    else
16.    {
17.        temp    =    car_straight_dias;
18.    }
19.
20.    car_straight_dias_old    =    car_straight_dias;
21.    car_straight_dias    =    temp;
22.    if    (fabs(car_center_dias)    <    10)
23.    {
24.        car_center_dias    =    0;
25.    }
26. }

```