

# 智能车黑线识别算法及控制策略研究

时间: 2009-05-18 11:23:07 来源: [电子技术](#) 作者: 北京信息科技大学, 机电工程学院 张淑谦 王国权

## 0 引言

“飞思卡尔”杯全国大学生智能车大赛是由摩托罗拉旗下飞思卡尔公司赞助由高等学校自动化专业教学指导委员会负责主办的全国性的赛事,旨在加强大学生的创新意识、团队合作精神 and 培养学生的创新能力。此项赛事专业知识涉及控制、模式识别、传感技术、汽车电子、电气、计算机、机械等多个学科,对学生的知识融合和动手能力的培养,对高等学校控制及汽车电子学科学术水平的提高,具有良好的推动作用。

智能车竞赛所使用的车模是一款带有差速器的后轮驱动模型赛车,它由大赛组委会统一提供。自动控制器是以飞思卡尔16位微控制器 MC9S12DG128 (S12) 为核心控制单元,配合有传感器、电机、舵机、电池以及相应的驱动电路,它能够自主识别路径,控制车高速稳定运行在跑道上。比赛要求自己设计控制系统及自行确定控制策略,在规定的赛道上以比赛完成的时间短者为优胜者。赛道由白色底板和黑色的指引线组成。根据赛道的特点,比赛组委会确定了两种寻线方案: 1. 光电传感器。2. 摄像头。

两种寻线方案的特点如下:

(1) 光电传感器方案。通过红外发射管发射红外线光照射跑道,跑道表面与中心指引线具有不同的反射强度,利用红外接收管可以检测到这些信息。此方案简单易行程序调试也简单且成本低廉,但是它受到竞赛规则的一些限制(组委会要求传感器数量不超过16个(红外传感器的每对发射与接收单元计为一个传感器, CCD 传感器计为1个传感器)),传感器的数量不可能安放的太多,因而道路检测的精度较低,能得到指引线的信息量也较少。若采用此方案容易引起舵机的回摆走蛇形路线。

(2) 摄像头方案。根据赛道的特点用黑白图像传感器即可满足要求。CCD 摄像头有面阵和线阵两种类型,它们在接口电路、输出信号以及检测信息等方面有着较大的区别,面阵摄像头可以获取前方赛道的图像信息,而线阵 CCD 只能获取赛道一条直线上的图像信息。摄像头方案的所能探测的道路信息量远大于光电传感器方案,而且摄像头也可以探测足够远的距离以方便控制器对前方道路进行预判。虽然此方案对控制器的要求比较高,但组委会提供的 MC9S12DG128 (S12) 的运算能力以及自身 AD 口的采样速度完全能够满足摄像头的视频采样和大量图像数据的处理的要求。

本文就是在摄像头方案的前提下,在实时的图像数据获取的基础上对图像信息进行数据处理,从而提取赛道中心的黑色指引线,再以此来作为舵机和驱动电机的控制依据。

## 1 摄像头采样数据的特点

采用的黑白摄像头的主要工作原理为:按一定的分辨率,以隔行扫描的方式采集图像上的点,当扫描到某点时,就通过图像传感芯片将该点处图像的灰度转换成与灰度一一对应的电压值,然后将此电压值通过视频信号端输出,见图1。摄像头连续地扫描图像上的一行,则输出就是一段连续的电压信号,该电压信号的高低起伏反映了该行图像的灰度变化。当扫描完一行,视频信号端就输出一个低于最低视频信号电压的电平(如 0.3V),并保持一段时间。这样相当于紧接着每行图像信号之后会有一个电压“凹槽”,此“凹槽”叫做行同步脉

冲，它是扫描换行的标志。然后，跳过一行后(因为摄像头是隔行扫描的)，开始扫描新的一行，如此下去，直到扫描完该场的视频信号，接着又会出现一段场消隐区。该区中有若干个复合消隐脉冲，其中有个远宽于(即持续时间长于)其它的消隐脉冲，称为场同步脉冲，它是扫描换场的标志。场同步脉冲标志着新的一场的到来，不过，场消隐区恰好跨在上一场的结尾和下一场的开始部分，到该场消隐区过去，下一场的视频信号才真正到来。摄像头每秒扫描25幅图像，每幅又分奇、偶两场，先奇场后偶场，故每秒扫描50场图像。奇场时只扫描图像中的奇数行，偶场时则只扫描偶数行。

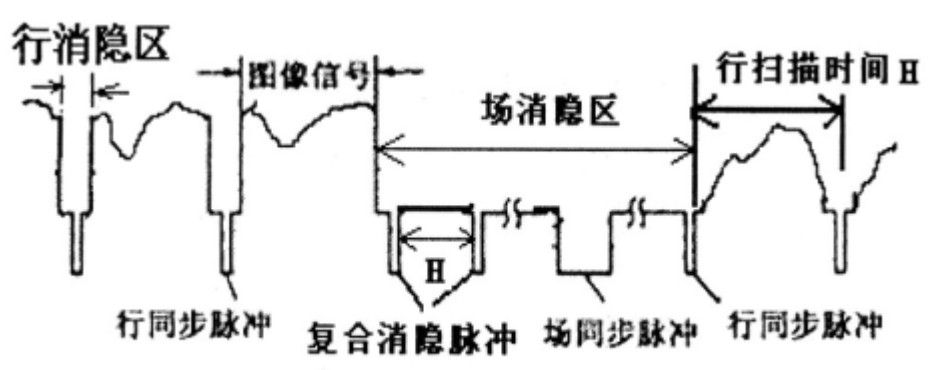


图1 摄像头的视频信号

因MC9S12DG128(S12)控制芯片的处理能力达不到计算机的运算速度，故采用了目前市面上分辨率最小的320X240的CMOS单板摄像头，它同样可以每秒扫描501幅图像。为了扩大摄像头的扫描距离来满足图像和控制处理的需要，将S12单片机的总线周期超频到32M，这样摄像头每行最多能够采集70个点。

通过实验测得摄像头扫描的每场图像中有320个行信号，其中的第23行到第310行是视频信号，即每场中有用的视频信号有288行。第311行到下一场第22行是场消隐信号。在实际的图像处理和控制过程当中完全没有必要对全部288行视频信号进行处理。由于每行能采集到80个有点像素点，所以在每场视频信号中每隔15行对行信号采集一次，用18行视频信号来判断前方是直道还是弯道，从而对小车进行速度和转向的控制。

## 2 黑色指引线的提取算法

赛道是由白色KT板上铺设黑色指引线组成，外界干扰比较少容易提取黑线。据此采用的是图像处理方法是较为简单的边缘提取算法。边缘与图像的性质之所以能联系在一起，是由于图像中的物体与物体，或者物体与背景之间的交界可以视为边缘，图像的灰度及颜色急剧变化的地方也可以看作是边缘。在图像中有颜色的变化必定会有灰度的变化，故对于边缘的检测与提取焦点集中在灰度上就可以。由于黑色指引线和白色跑道的色差较大，可以确定一个分别黑白颜色的阈值，通过实验确定该阈值(VALUE)为8。由于光照强弱的原因黑色像素的灰度值可能不完全一样，实际操作中我们发现只要灰度值小于4(DIFFVAL)的点即可视为黑色点。

摄像头扫描的图像分为一下两种情况：

(1)扫描图像只有一条竖直黑线，如图2所示



图2 只扫描到竖直黑线

从左边的第一个有效像素点(useful\_line)开始依次对相邻的像素点进行灰度值相减(左减右),当两者的差值大于阈值(VALUE)时,判定下一个像素点(usful\_line+1)对应的为黑线的左边缘。继续相减当差值再次大于阈值时,将此像素点定位黑线的右边缘。该程序如下所示:

```

unsigned char AbsSub(unsigned char n1,
unsigned char n2)
{ unsigned char dif = 0;
  if(n1 > n2)
  { dif = n1 - n2;
  }
  else
  { dif = n2 - n1;
  } return dif;
} //程序的功能为取两数之差的绝对值,是黑
线算法的准备程序
for(us_row = 0; us_row < ROW_MAX; us_row++)
  //对采集到图像的每行进行扫描
  {
    for(us_line = 0; us_line < LINE_MAX-1;
us_line++) //确定黑线左边缘
    { g_ucDif =
AbsSub(ucImage_data[us_row][us_line],
ucImage_temp);
      if(g_ucDif <= DIFFVAL)
      { left = us_line;
        right = LINE_MAX-1;
        for(linel = left; linel < LINE_MAX-1;
linel++) //确定黑线右边缘
        {
          g_ucDif=AbsSub(ucImage_data[us_row][left],ucImage
_data[us_row][linel+1]);
          if(g_ucDif > VALUE)
          { right = linel;
            break;
          }
        }
      }
    }
  }
}

```

其中 g\_ucDif 为两个像素灰度值之差的绝对值, LINE\_MAX 为采集图像的最大有效列数, ROW\_MAX 为采集图像的最大有效行数, ucImage\_data[][]为存储图像像素点的一个二维数组, ucImage\_temp 为临时存储图像的数组。

(2)扫描到的图像为黑色交叉线,如图3所示



图3 扫描到黑色交叉线

前面提到 S12 单片机的总线超频以后通过 AD 转换每一行图像我们能采集到 70 个有效数据点，而黑线的宽度只有 25mm (大赛组委会规定)，在这个宽度内最能只能包含 6 个有效的数据点。因此对于黑色交叉线我们只要将右边缘(right)所在列数与左边缘(left)所在列数相减，如果差值大于 6，我们即可判定此时为黑色交叉线。程序只要在前面的基础上加上下语句即可完成。

```
if(right - left > 6)
//黑色交叉线
{
    Blackwire_data[us_row]=(LINE_MAX-1)/2;
}
break;
```

其中 Blackwire\_data[] 为采集到的黑线中心线数据，它将在后面的控制策略中被用到。

### 3 智能车的控制算法

在比赛中，影响赛车速度的一个关键的因素就是赛车能不能提前识别弯道与直道，也就是说赛车的控制算法对提高赛车的性能起到至关重要的作用。这里采用鲁棒性好且算法简单的经典 PID 算法。但普通的 PID 是线性算法，若参数设置太大则小车在走直道时很容易出现振荡的情形，而在高速走弯道的时则易出现反应迟钝、冲出赛道的情况。因此采用非线性的 PID 算法，使之对小偏差不敏感而对大偏差敏感，实现了直道平稳同时弯道灵敏的目的。

驱动电机的 PID 控制部分通过下面算法实现：

$$MV_n = MV_{n-1} + \Delta MV_n$$

$$\Delta MV_n = K_p(e_n - e_{n-1}) + K_i e_n + K_d((e_n - e_{n-1}) - (e_{n-1} - e_{n-2}))$$

其中， $MV_n$ 、 $MV_{n-1}$ ：当前和上次操作量； $\Delta MV_n$ ：当前操作量微分； $e_n$ 、 $e_{n-1}$ 、 $e_{n-2}$ ：当前，上次，上上次偏差； $K_p$ 、 $K_i$ 、 $K_d$  的数值通过实验得到。

经过测试发现，对于舵机控制当它的积分环节  $k_i$  为 0 时，小车的表现性能要更好，所以对舵机采用的是带非线性的 PD 算法。

该算法首先需要知道所采集图像中黑线的转弯斜率(如图4所示)，图中直线表示赛道为直道，左右两边的线分别表示左转弯和右转弯时摄像头可能采集到的黑线情形，不同的线表示不同的转弯斜率(slope)，在后面的 PD 控制中我们将用到此斜率。

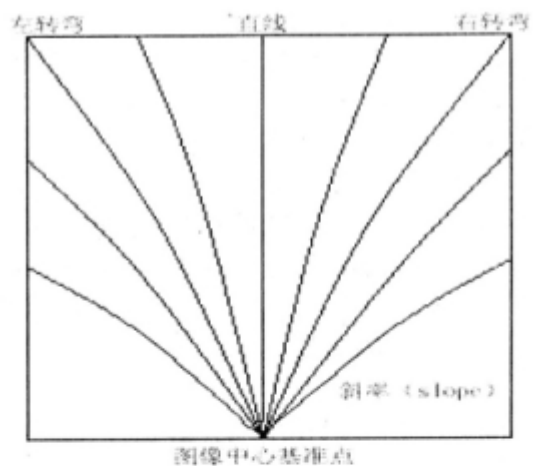


图4 黑线转弯斜率示意图

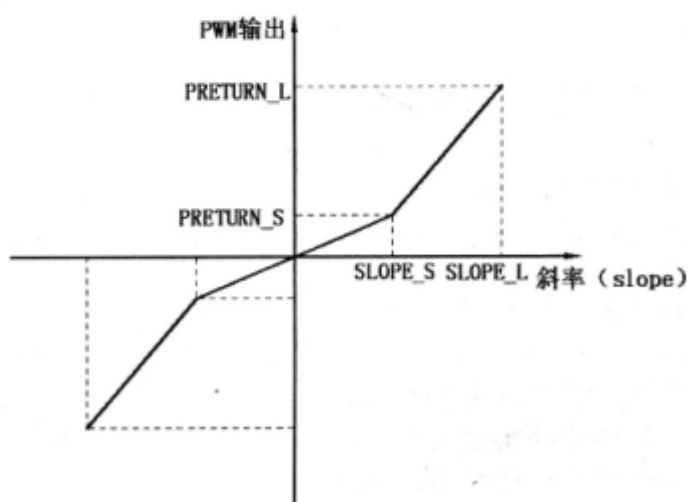


图5 PD非线性输出示意图

采用的是带有预判的控制策略(如图5所示)，具体分为以下几种情况：

(1)赛道一直为直道时，程序如下：

```

if (当前图像==直道)
{
    pwm=STEER_COEFFICIENT(1)* g_sCurE;
}
其中STEER_COEFFICIENT(x) =

$$\frac{\text{STEER\_RIGHT\_LIMIT} - \text{STEER\_LEFT\_LIMIT}}{\text{LINE\_MAX}}(x)$$

g_sCurE为当前的转弯偏差。并且STEER_LEFT_
LIMIT, STEER_RIGHT_LIMIT 为舵机的左、右转弯极限。
(2) 赛道为直道加弯道时, 程序如下:
if(当前图像==直道 && 下一幅图像==左转弯道)
{
    pwm=((PRETURN_L-PRETURN_S)*slope+
    SteerSlope_bcSUBad)/(SLOPE_L-SLOPE_S);
    return pwm;
} else if(当前图像==直道 && 下一幅图像==右
转弯道)
{
    pwm=0-(((PRETURN_L-PRETURN_S)*slope+
    SteerSlope_bcSUBad) / (SLOPE_L-SLOPE_S));
    return pwm;
}
其中, SteerSlope_bcSUBad =  $\frac{\text{PRETURN\_S} \cdot \text{SLOPE\_L}}{\text{SLOPE\_S} \cdot \text{PRETURN\_L}}$ 
(3) 赛道为连续弯道(S弯)时, 程序如下:
if(当前图像==右转弯道&&下一幅图像==左转弯道)
{
    if(左转弯道)
    {
        pwm=g_10ldPwm+STEER_COEFFICIENT(1) *
        g_sCurE+kd * (g_sCurE-g_sOldE);

        } else if(右转弯道)
        {
            pwm=STEER_CENTER+STEER_COEFFICIENT(1)
            * g_sCurE;
        } else if(当前图像==右转弯道 && 下一幅
        图像==右转弯道) {
            pwm=g_10ldPwm+STEER_COEFFICIENT(1) *
            g_sCurE+kd * (g_sCurE-g_sOldE);
        }
        if(当前图像==左转弯道 && 下一幅图像==左转弯
        道)
        {
            pwm=g_10ldPwm+STEER_COEFFICIENT(1) *
            g_sCurE + kd * (g_sCurE - g_sOldE);
        } else if(当前图像==左转弯道 && 下一幅图像
        ==右转弯道)
        {
            if(左转弯道)
            {
                pwm=g_10ldPwm+STEER_COEFFICIENT(1) *
                g_sCurE+kd * (g_sCurE-g_sOldE);
            } else if(右转弯道)
            {
                pwm=STEER_CENTER+STEER_COEFFICIENT(1) *
                g_sCurE;
            }
        }
    }
}

```

其中, STEER\_CENTER 为舵机转向中心, g\_10ldPwm 为前一步的 PWM 输出, g\_sCurE 为当前的转弯偏差, g\_sOldE 为前一步的转弯偏差。

#### 4 结论

本文在摄像头已经能够采集较为准确图像的基础上, 提出并研究了黑色指引线的边缘提取算法, 然后在该算法的基础上决定了舵机转向的控制策略, 通过实践证明, 采用此种算法和控制策略可以使赛车获得较快的行驶速度和较为可靠的转向性能。