



哈爾濱工業大學(深圳)
Harbin Institute of Technology Shenzhen

数字图像处理课程设计报告

学院： 机电工程与自动化学院

题目： 自动停车

班级： 自动化二班

姓名： 朱方程、彭沛

学号： SZ170410221, SZ170710122

教师： 陈浩耀、葛亚明

上交日期： 2018. 12. 11

目 录

1 引言	3
1.1 编写目的.....	3
1.2 背景.....	3
2 任务概述.....	3
3 系统设计与代码.....	4
3.1 主程序设计.....	4
3.2 停车函数设计.....	5
3.2 代码结构（流程图）	6
3.4 拓展部分设计.....	6
4 调试与测试.....	7
4.1 调试过程.....	7
4.2 测试结果.....	7
5 小组分工.....	7
6 编程中遇到的问题.....	8
6.1 问题 1.....	8
6.2 问题 2.....	9
6.3 问题 3.....	10
7 分析总结与心得体会.....	10
8 完整代码.....	11

1 引言

1.1 编写目的

在为期几天的编写代码、调试分析过程中，小组成员有过一些灵光乍现的时刻，产生了一些看起来不错的点子，但在实现过程中也产生了各种各样的问题。

为了实现小车自动识别停车标志“H”，并自动导航到指定停车位置的目标，小组成员积极查找资料、寻找思路、编写代码、调试分析。为了将这个过程中产生的问题以及相应解决措施、编写代码的思路以及收获等记录下来，为以后从事相关研究提供参考资料，故编写此课程设计报告。

1.2 背景

数字图像结课，老师为了让同学们将理论知识用于实际开发，巩固掌握所学内容，故布置三个题目，大家自由选择并完成。

本小组商讨决定，选择题目一“Parking”：识别停车标志——字母“H”，并自动行驶到车标前方或正上方，实现自动停车的目标。

2 任务概述

Parking: 识别停车标志——绿色字母“H”，并自动行驶到车标前方或正上方，实现自动停车的目标。

停车标志如下：



3 系统设计与代码

3.1 主程序设计

运动控制采用比例控制。

先获取左目摄像头图像，考虑到图像颜色受光照影响很大，加入调节图像亮度和对比度的函数。之后将图像由 RGB 颜色模型转到 HSV 颜色模型，分割绿色，生成二值化图像，统计整个图像亮点和暗点个数并打印。判断本帧图像亮点数是否小于阈值且上一帧图像亮点数大于阈值。

若是，则小车以 0.15 的速度行驶两秒后停止。若否，则将图像沿水平方向均匀分成五部分左 1、左 2、中间、右 1、右 2，分别统计五个区域中亮点个数、占总亮点数的比例。

根据五部分的亮点所占比例设置偏转角，若左 1 区域大于 0.5 且右 1 区域小于 0.01，则偏转角设为 3；否则，若左 1 大于 0.15 且右 1 小于 0.01，则偏转角设为 2；否则，若左 2 大于 0.5 且右 2 小于 0.01，则偏转角设为 2；否则，若左 2 大于 0.15 且右 2 小于 0.01，则偏转角设为 1；否则，若右 2 大于 0.5 且左 2 小于 0.01，则偏转角设为-3；否则，若右 2 大于 0.15 且左 2 小于 0.01，则偏转角设为-2；否则，若右 1 大于 0.5 且左 1 小于 0.01，则偏转角设为-2；否则，若右 1 大于 0.15 且左 1 小于 0.01，则偏转角设为-1；否则，偏转角设为 0。

```
double vel_angle;
if ((left1_pro >= 0.5) && (right1_pro <= 0.01) && (vel_forward != 0))
    vel_angle = 3;
else if ((left1_pro >= 0.15) && (right1_pro < 0.01) && (vel_forward != 0))
    vel_angle = 2;
else if ((left2_pro >= 0.5) && (right2_pro < 0.01) && (vel_forward != 0))
    vel_angle = 2;
else if ((left2_pro >= 0.15) && (right2_pro < 0.01) && (vel_forward != 0))
    vel_angle = 1;
else if ((right1_pro >= 0.15) && (left1_pro < 0.01) && (vel_forward != 0))
    vel_angle = -1;
else if ((right1_pro >= 0.5) && (left1_pro < 0.01) && (vel_forward != 0))
    vel_angle = -2;
else if ((right2_pro >= 0.15) && (left2_pro < 0.01) && (vel_forward != 0))
    vel_angle = -2;
else if ((right2_pro >= 0.5) && (left2_pro < 0.01) && (vel_forward != 0))
    vel_angle = -3;
else
    vel_angle = 0;
```

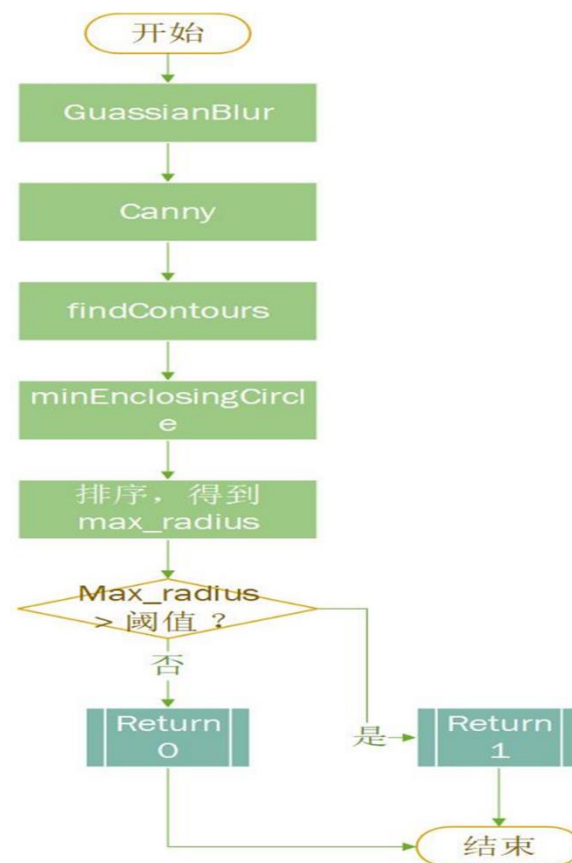
3.2 停车程序设计

当 H 竖直放置时，用停车函数判断其是否停车。输入二值化图像和阈值，返回 0 或 1，0 表示不停车，1 表示停车。

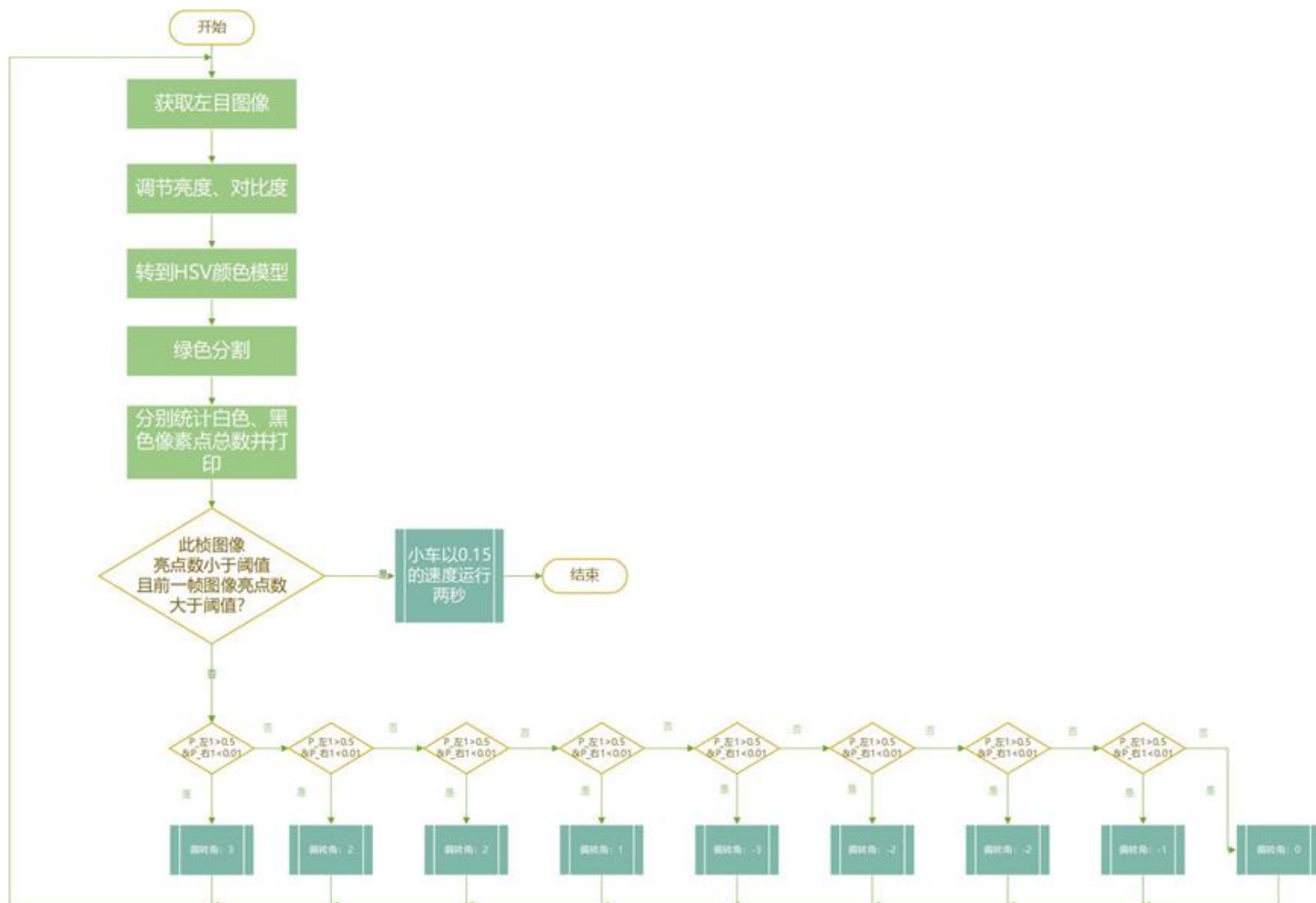
思路一：统计亮点的个数，大于阈值则停止。因觉此无何难度，便无具体实现。

思路二：计算“H”的最小外接圆半径，大于阈值则停止。以下为其具体实现：

先把图像用高斯模糊平滑处理，接着用边缘检测函数检测边界，再用寻找轮廓函数寻找图像轮廓，再用获取最小外接圆函数对轮廓寻找最小外接圆。为消除噪声影响，对得到的最小外接圆半径排序，此处取半径最大的最小外接圆，再在图上标注出来。如果最大的最小外接圆半径大于阈值，则返回 1，否则返回 0。



3.3 代码结构（流程图）



3.4 拓展部分设计

对于正对着“H”正前方进入的情况，有具体的实现思路，但因其过于复杂，在此只做了部分尝试，未完全完成。

思路：

先将图像进行仿射变换，将不平行的“H”转换成“横平竖直”的“H”，再将图像用高斯模糊平滑处理，接着用边缘检测函数检测边界，再用寻找轮廓函数寻找图像轮廓，再用获取最小外接斜矩形函数对轮廓寻找最小外接斜矩形。为消除噪声影响，对得到的最小外接斜矩形的对角线长度排序，此处取对角线最长的最小外接斜矩形，然后在图上标注出来。最后可根据最小外接斜矩形函数返回的旋转角度对小车前进方向进行修正。

4 调试与测试

4.1 调试过程

实验中需要调试的参数主要有：

- i. 小车的运动控制参数：速度、转角；
- ii. 亮点数小于阈值后的滑行时间；
- iii. 颜色分割函数 `inRange`、亮度、对比度调整函数 `convertTo`；
- iv. 各类阈值：亮点数阈值、停车函数最小外接圆半径阈值；

4.2 测试结果

- i. 运动参数：由于仅采用比例控制，参数大时效果不理想，速度一般取 ≤ 0.2 ，转角取 $\text{angle} \times 0.2$ ($\text{angle} = 3, 2, 1, 0, -3, -2, -1$)；
- ii. 亮点数小于阈值后的继续运行时间：以速度 0.15 运行 2 秒；
- iii. `inRange` 参数取标准绿色范围(`Scalar(43,43,46)`, `Scalar(77,255,255)`)、`convertTo` 函数当平放时光照充足，不需要调整也能有较好效果，立着时取 5(对比度)、50(亮度)；
- iv. 亮点数阈值取 75，半径阈值取 30；

5 小组分工

朱方程：负责主体代码框架的确定

- ① 利用 HSV 格式的图像通过颜色分割来识别停车标志 H。
- ② 完成行驶时的线速度和角加速度控制函数的初步设计。

利用比例 P 控制：将摄像头捕捉的图像区域的像素按列数分为左、中、右三个区域，计算识别到的 H 的像素点在三个区域所占的比例。根据左、中、右三部分有效像素所占比例的情况来进行控制。

```
double vel_forward;
if (Num[255]>75)
    vel_forward=1;
else
    vel_forward=0;

double vel_angle;
if ((left_pro>=0.5)&&(right_pro<=0.01)&&(vel_forward!=0))
    vel_angle=2;
else if ((left_pro>=0.15)&&(right_pro<0.01)&&(vel_forward!=0))
    vel_angle=1;
else if ((right_pro>=0.15)&&(left_pro<0.01)&&(vel_forward!=0))
    vel_angle=-1;
else if ((right_pro>=0.5)&&(left_pro<0.01)&&(vel_forward!=0))
    vel_angle=-2;
else
    vel_angle=0;
```

③ 尝试利用透视变换，将平面视图转化为鸟瞰图。

```
//透视变换
Mat PerspectiveTrans(Mat src, Point2f* scrPoints, Point2f* dstPoints)
{
    Mat dst;
    Mat Trans = getPerspectiveTransform(scrPoints, dstPoints);
    warpPerspective(src, dst, Trans, Size(src.cols, src.rows), CV_INTER_CUBIC);
    return dst;
}
```

但因难度较大，效果不理想最终没有投入使用。原因见第 6 部分“编程中遇到的问题”。

④ 课程设计报告的撰写。

彭沛：负责主体代码的优化和拓展：

① 优化行驶时线速度和角速度的控制函数：

将三部分细分成五部分，调试临界比例值；

② “H”立着放时的停车函数：

检测二值化图像的轮廓、获取“H”的最大的最小外接圆半径、大于阈值时返回 1，示意停止；

③ 尝试让车正对着“H”进去：

透视变换转换“H”，findContours 寻找轮廓，minAreaRect 函数获取最大的最小外接斜矩形，RotatedRect 类 minRect[].angle 返回矩形的旋转角度。实际情况远比理论复杂，其余的因能力有限，未能成功。

6 编程中遇到的问题

6.1 问题一

透视变换，将平面视图转化为鸟瞰图，实现难度较大，且小车距离停车标志较远时效果不理想。原因分析：

透视变换函数 warpPerspective 需要一个坐标变换矩阵来对图像进行透视变换，此坐标变换矩阵可由 getPerspectiveTransform 产生。产生方法是选定输入图像上四个点，并给出此四点在输出图像上对应的位置，从而产生一个有效的变换矩阵。因 H 所在的正方形图像是在地面上平铺，假设摄像头正对正方形，平视的图像应为上边短、下边长的一个等腰梯形，如下图(a)所示。

此时，可选等腰梯形四个顶点作为四个输入参考点，如下图中所选的输入参考点的坐标分别为(200,200), (400,200), (0,400), (600,400)，对应到输出图像中的位置为(50,0), (550,0), (0,400),(600,400)，得到的透视变换效果图如下图(b)所示。但因为小车在移动过程中，不可能保持始终正对“H”所在的正方形，且摄像头读入的“H”图像在视野中的位置不断变化，故提供给 `getPerspectiveTransform` 函数的四个参考点的位置也要相应地改变，但四个点的位置很难用一个普适的方法来确定，故没有投入使用。

```
Point2f AffinePoints0[4] = { Point2f(200, 200), Point2f(400, 200), Point2f(0, 400), Point2f(600, 400) };
```



Figure(a)



Figure(b)

6.2 问题二

当摄像头捕捉不到停车标志“H”时，按照本程序的控制思路，小车立即停止前进。但常常此时小车距离“H”仍有一段距离，不能达到停在“H”前方的目标。

解决办法：当满足停止条件时，让小车以一定的线速度前进一定时间后停止，即可停在指定位置，不会离“H”标志太远。

```
//若满足停止条件，则滑行一段距离到达停车标志前方或上方
if (vel_forward == 0 && vel_pre != 0)
{
    for (i = 0; i < 1000; i++) {
        cmd_red.linear.x = 0.15;
        pub.publish(cmd_red);
        waitKey(2);
    }
}
vel_pre = vel_forward;
```

6.3 问题三

“H”立着时，光照不够，绿色变成黑色，得到的“H”不理想；
可通过 `convertTo` 函数调整图像的对比度和亮度，相当于手动补光：

```
1. alpha = 50;
2. beta = 5;
3. frIn.convertTo(frIn, frIn.type(), alpha, beta);    //亮度对比度调节
```

6.4 问题四

“H”的轮廓里在不受外界噪声影响时也存在不止一个的最小外接圆半径、最小外接斜矩形对角线长度。

排序取最大值，同时可消除噪声的影响；

7 分析总结与心得体会

一开始觉得课程设计的要求太高，以自己目前的水平几乎不可能完成。但通过冷静分析、查阅资料、与队友讨论交流，以及不断地尝试不同的实现方法与思路，不断地调试各项参数，不断地 `debug`，最终还是用一些简单的算法实现了目标功能。在这个过程中，收获颇丰，受益匪浅。

十分感谢老师的辛勤付出以及队友的支持！