

# 基于 AI 视觉的移动机器人自动驾驶系统设计

王 茜, 李振华

(北京信息科技大学 信息与通信工程学院, 北京 102206)

**摘要:** 针对移动机器人的自动驾驶问题, 提出了一种以树莓派为核心的基于视觉的自动驾驶控制系统设计方案。通过解析基于麦克纳姆轮的全向移动机器人运动模型, 设计了移动机器人的硬件控制系统。通过摄像头采集道路图像, 利用 OpenCV 技术获取道路图像信息。根据树莓派图像处理的数据, 采用 PID 算法控制机器人移动, 使其能按照指定轨迹行驶。结果表明: 利用该系统, 移动机器人能自主识别路径并实现自动驾驶。

**关键词:** 自动驾驶; OpenCV 技术; 图像识别; PID 算法; 自主识别路径; 麦克纳姆轮

**中图分类号:** TP242.6

**文献标识码:** A

**文章编号:** 2095-1302 (2024) 10-0147-04

## 0 引言

随着人工智能时代的到来, 人们的交通方式也开始发生改变, 各种交通工具不断升级, 逐渐变得智能化。从自行车、汽车到高铁、飞机, 再到共享单车、共享汽车、无人驾驶汽车, 等等, 交通方式的变化得益于科技的迅速发展, 这些智能交通工具大大便利了我们的出行, 提升了生活的幸福感。

当前, 无人驾驶汽车在交通发展领域备受重视, 而自动驾驶技术是无人汽车的核心。引导方法是目前应用最多的导航方式, 即机器人对感知的某些外部连续路径参照线作出相应反应来进行导航<sup>[1]</sup>。比如, 在地下埋引导线等。但该方法在维修成本、可靠性、抗干扰能力等方面难以满足用户需求。近年来, 视觉方法被广泛认为是一种更合适的导航方式, 当路面上有路径引导线时, 机器人就能通过视觉系统按照指定路线行进。本文设计了一种移动机器人, 其行进功能主要由 4 个电机驱动的麦克纳姆轮来实现, 通过摄像头采集路径信息, 利用 OpenCV 技术进行图像处理, 并通过 PID 控制算法计算移动参数, 将参数提供给底盘运动控制系统, 实现机器人的前行或转弯。

## 1 移动机器人的运动原理分析

机器人采用麦克纳姆轮作为基础原件, 实现 360° 全向运动。麦克纳姆轮由轮和围绕轮的辊子组成, 辊子轴线和轮的轴线夹角是 45°。麦克纳姆轮有互为镜像关系的 A、B 轮, A 轮向前运动时同时向右运动, 即斜向右前方运动, 向后运动的同时会向左运动, 即斜向左后方运动; 相应 B 轮就可以斜向左前和右后方运动。如图 1 所示, 4 个麦克纳姆轮位于机器人正方形底座的 4 个顶点, 本设计采用 ABBA 的分布方

式, 移动平台固定安装 4 个直流电机分别驱动 4 个轮子, 机器人能够实现前后运动、水平移动、绕中心自转等功能<sup>[2]</sup>。

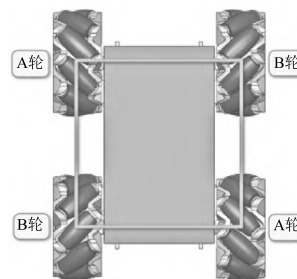


图 1 麦克纳姆轮分布图

图 2 对单个麦克纳姆轮进行运动学分析,  $R$  表示轮的半径, 实际辊子长度为  $l_0$ 。  $V_{ix}$  为第  $i$  轮沿  $X$  轴方向的分速度,  $V_{iy}$  为第  $i$  轮沿  $Y$  轴方向上的分速度,  $V_g$  为辊子的速度, 辊子轴线与全向轮轴线夹角为  $\alpha$ ,  $\omega_i$  为全向轮绕轮轴的转速,  $i=1, 2, 3, 4$ , 分别代表了左前轮、右前轮、左后轮、右后轮, 4 个轮的转速和移动平台的运动呈一一对应关系。本设计通过调节各轮的转速, 实现了移动平台的不同运动状态<sup>[3]</sup>, 公式如下:

$$\begin{bmatrix} V_y \\ V_x \\ \omega \end{bmatrix} = \frac{R}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \tan 45^\circ & \tan 45^\circ & \tan 45^\circ & \tan 45^\circ \\ -\frac{1}{l_0} & \frac{1}{l_0} & -\frac{1}{l_0} & \frac{1}{l_0} \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} \quad (1)$$

## 2 移动机器人控制系统设计

移动机器人最重要的组成部分是控制系统。框架采用亚克力板制作, 控制系统采用分层结构设计: 最上层安装摄像头, 用于实时采集图像; 中间层安装树莓派控制板, 负责处理图像, 通过串口与摄像头连接; 底层安装下位机 STM32 控制板, 通过串口与树莓派连接。此外, 还包括为整个系统供电的电源模块、促使机器人移动的电机驱动模块等。控制

系统总体结构如图3所示。

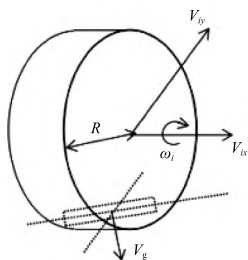


图2 单轮运动示意图

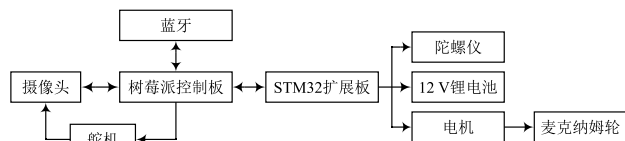


图3 控制系统总体结构

上位机采用树莓派进行图像处理，并给下位机传输运动指令。下位机通过STM32驱动机器人的底盘并实现其他硬件的信号处理。

图像采集模块：摄像头模块在PWM舵机的控制下，进行三自由度道路图像信息采集。

图像处理模块：主控芯片实时接收并分析图像信息，获得图像中路径引导线的位置坐标，通过计算将数据传输至下位机。

驱动控制模块：STM32控制板接收程序并分析上位机下发的指令，控制电机驱动机器人行进；同时采集移动平台上搭载的相关模块状态及传感器数据并上传至树莓派<sup>[4]</sup>。编码器用来测量机器人4个轮子的旋转速度，并反馈测量值到单片机，实现机器人运动速度的闭环控制。MPU6050陀螺仪负责采集机器人的偏航角度<sup>[5]</sup>。

电源模块：通过18650电池组为STM32供电，再通过12V转5V模块为系统各功能模块供电。

### 3 移动机器人视觉导航系统设计

移动机器人在自动驾驶过程中，最重要的是道路引导线识别。本设计采用树莓派原装Raspberry Pi Camera模块进行图像采集，其具备500万像素的OV5647镜头，静态图片分辨率为 $2592 \times 1944$ ，编程语言选择Python，在Jupyter Lab中实现编程<sup>[6]</sup>。

如图4所示，系统首先对原始输入图像进行透视变换，然后根据设定的阈值二值化图像挑选出目标点（即路径中的引导线）并消除其中的噪声点，从而构成目标的支持点集；路线辨识模块可以通过目标支持点集来检查工作场景中的引导线，得到所需要的路线信号；根据机器人路径走向偏离中线角度的大小，分直行和左右转弯两类，通过调整麦克纳姆

轮的转速和转向，实现机器人的路径跟踪<sup>[7]</sup>。路线识别具体步骤如下：

(1) 建立OpenCV坐标系，通过图片读取函数获取图像，并设定图像尺寸为 $320 \times 240$ 。利用OpenCV提供的投影变换函数实现透视变换，把车身前的区域变换成俯视图。

(2) 为了降低图像处理过程中的噪声影响，引入改进的中值滤波器算法。算法可有效降低中值滤波的时间复杂度，达到对图像进行实时处理的要求<sup>[8]</sup>。再对去除噪点的图像进行二值化处理，可以创建点集直方图。

(3) 取直方图数组左半部分中最大值的坐标位置为左线，同理为右线。利用数据确定黑线的位置，并绘制两条黑线，之后计算图像中线和实际中线的位置。

(4) 找到黑线后，返回实际中线和图像中线的差值。

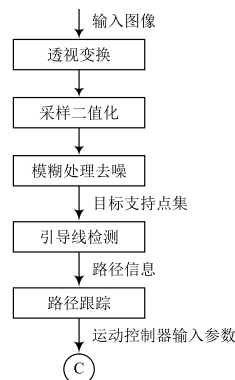


图4 视觉导航工作流程

### 4 移动控制算法设计

目前，PID算法被广泛应用于生活和工业生产中。PID控制算法基本原理如图5所示，对输入偏差进行比例积分、微分运算，利用运算的叠加结果控制执行机构。比例环节可对偏差做出及时响应，积分环节用于消除静态误差，微分环节可阻止偏差的变化。图5中： $u(t)$ 为PID调节器的输出； $e(t)$ 为PID调节器的偏差，即设定值 $r(t)$ 与实际测量值 $y(t)$ 的偏差<sup>[9]</sup>。

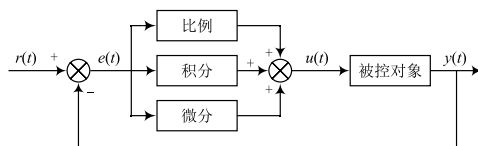


图5 PID控制算法原理

数字PID控制算法可以分为位置式PID和增量式PID控制算法。本设计采用位置式PID控制算法，其输出是被控对象的实际控制量。算法表达式如下：

$$u_k = K_p \cdot e_k + K_i \sum_{j=0}^k e_j + K_d (e_k - e_{k-1}) \quad (2)$$

由式(2)可知，位置式算法的积分项要用到偏差的累

加值,容易产生较大的累加误差,也会导致系统的运算量变大。为提高稳定性,改善瞬态响应<sup>[10]</sup>,令 $P=0.85$ , $I=0$ , $D=0.3$ ,具体控制代码如下:

```
class PositionalPID:
```

```
def __init__(self, P, I, D):
```

```
    self.Kp = P
```

```
    self.Ki = I
```

```
    self.Kd = D
```

```
    self.SystemOutput = 0.0
```

```
    self.ResultValueBack = 0.0
```

```
    self.PidOutput = 0.0
```

```
    self.PIDErrADD = 0.0
```

```
    self.ErrBack = 0.0
```

设置 PID 控制器参数:

```
def SetStepSignal(self, StepSignal):
```

```
    Err = StepSignal - self.SystemOutput
```

```
    KpWork = self.Kp * Err
```

```
    KiWork = self.Ki * self.PIDErrADD
```

```
    KdWork = self.Kd * (Err - self.ErrBack)
```

```
    self.PidOutput = KpWork + KiWork + KdWork
```

```
    self.PIDErrADD += Err
```

```
    self.ErrBack = Err
```

设置一阶惯性环节系统 (InertiaTime 为惯性时间常数):

```
def SetInertiaTime(self, InertiaTime, SampleTime):
```

```
    self.SystemOutput = (InertiaTime * self.ResultValueBack + \
        SampleTime * self.PidOutput) / (SampleTime + \
```

```
InertiaTime)
```

```
    self.ResultValueBack = self.SystemOutput
```

通过视觉图像处理计算的偏移量 Bias 即为 PID 出口值,将偏移量限制在  $-20 \sim 20$ ,防止转向过快丢失赛道黑线。通过函数 `Z_axis_pid.SetInertiaTime(0.5, 0.2)` 配置惯性时间常数。最后通过均值滤波法找出点集直方图中的峰值个数,2 或 4 个峰值时认为车身偏移不大,直接利用机器人速度控制函数控制小车左右转向即可。1 个峰值时机器人即将跑出赛道,分为看见左线和看见右线两种情况,若  $Bias=-20$  实现右转,若  $Bias=20$  则实现左转,其他峰值时机器人直走。

## 5 测试结果及分析

### 5.1 测试环境

机器人移动测试场地如图 6 所示。场地中有两条 5 cm 宽的黑色引导线。为检验机器人的路线识别能力,在场地中设置了直线、曲线和拐角。在实验过程中,机器人未提前装载地图路线,全程跟随引导线行驶。实验路线边界清晰。

实验机器人采用麦克纳姆轮底盘作为移动平台,4 路直流电机负责驱动,安装并连接了 12.6 V 的电池模块到 STM32 控制板。

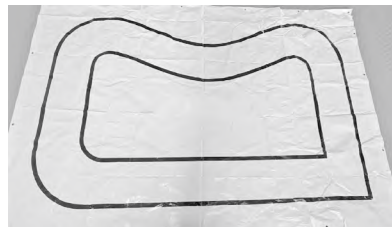
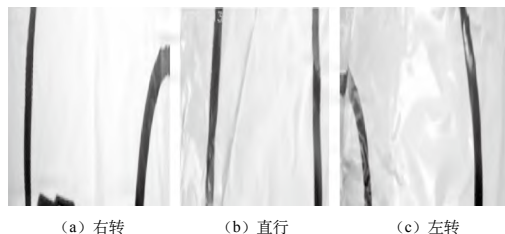


图 6 引导路线图

安装图像分辨率为  $2592 \times 1944$  的树莓派原装 OV5647 摄像头以采集视觉图像。通过 WiFi 连接手机,实时显示摄像头采集的路线图像。

### 5.2 引导线检测测试

移动机器人在行驶过程中,通过摄像头模块采集图像并对采集的图像进行处理,确定是否有引导线出现。本测试采用的引导线如图 6 所示。当检测到有引导线时,对该图像做进一步处理,获取引导线的位置坐标信息。由于机器人在不断行进,所以要求对图像检测快且准。对直行和弯道引导线进行测试,效果如图 7 所示。在引导线检测调试过程中,经过反复实验,发现灯光的反射还有地板颜色会在一定程度上影响图像二值化处理的准确性。

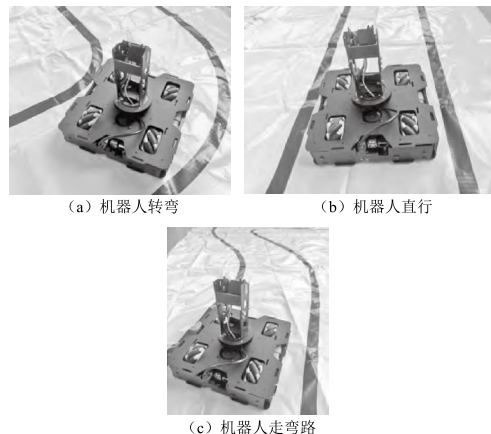


(a) 右转 (b) 直行 (c) 左转

图 7 机器人视野中的场景变化图像

### 5.3 自动驾驶测试

通过人为放置方式测试移动机器人的自动驾驶功能。放置路线后,通过 WiFi 连接机器人和手机,使机器人开启自动驾驶模式。为保证机器人行驶的准确性,在直行路线和弯道处进行多次测试,效果如图 8 所示。



(a) 机器人转弯 (b) 机器人直行

(c) 机器人走弯路

图 8 不同路线自动驾驶测试效果



在图6所示的模拟场景中进行50次模拟驾驶测试,结果见表1所列。

表1 指定路线自动驾驶测试结果

模拟场景	测试 次数/次	成功 次数/次	失败 次数/次	成功率/%
直行测试	50	50	0	100
转弯测试	50	50	0	100
曲线测试	50	50	0	100

#### 5.4 实验结果及分析

将机器人放在引导地图上,经过多次实验,其均能以较快的速度进行直走和转弯,并且通过多次测试,机器人行驶过程中均未超出引导线。实验表明,该机器人具有良好的自动驾驶功能。

引导线识别测试过程中,在外界环境良好的情况下机器人能够准确识别黑色引导线。在扫描到地图外侧地板的情况下,受颜色影响,机器人在识别黑色引导线与深色地板时会产生一定程度的偏差。自动驾驶测试过程中,移动机器人能够准确识别引导线,并按照路线成功直行和转弯。通过多次测试,证明在引导线内直行和转弯成功率均较高。

## 6 结 语

针对移动机器人的自动驾驶问题,利用OpenCV在树莓

派中设计了路线识别算法,以视觉方式实时获取两侧引导线与中线坐标;运用位置式PID控制算法,根据图像计算的中线与实际中线的偏差,控制调节机器人的转向角度。结果表明:该控制方法实现了基于视觉的机器人自动驾驶,并经过多次重复实验检测,证明了在简单直行道路和弯路中成功率很高,移动机器人可以在引导线内准确行驶。

## 参 考 文 献

- [1] 张海波,原魁,周庆瑞.基于路径识别的移动机器人视觉导航[J].中国图象图形学报,2004,9(7):89-93.
- [2] 王兴松.Mecanum轮全方位移动机器人技术及其应用[J].机械制造与自动化,2014,43(3):1-6.
- [3] 张忠民,郑仁辉.基于模糊PID的麦克纳姆轮移动平台的控制算法[J].应用科技,2017,44(6):53-59.
- [4] 封斌,闫志辉.基于Arduino的服务机器人控制系统设计与实现[J].广州航海学院学报,2018,26(1):63-66.
- [5] 李娟,秦伟.基于视觉的移动机器人避障控制系统设计[J].机床与液压,2021,49(15):24-28.
- [6] 韩宇,张磊,吴泽民,等.基于嵌入式树莓派和OpenCV的运动检测与跟踪系统[J].电视技术,2017,41(2):6-10.
- [7] 周南.基于图像识别的智能小车路径跟随系统设计[D].长沙:湖南大学,2017.
- [8] 杨春晖.基于地面轨线的单目机器人视觉导航系统设计[D].苏州:苏州大学,2010.
- [9] 朱峰涛,武洪涛.基于增量式PID算法的直流电机调速系统[J].仪表技术与传感器,2017(7):121-126.
- [10] 谢芳芳,杨鉴.单关节机器人伺服系统的建模与仿真[J].湖南工业职业技术学院学报,2021,21(1):47-52.

作者简介:王 茜(2003—),女,研究方向为物联网工程。

李振华(1977—),男,博士,讲师,研究方向为智慧感知与控制。

(上接第146页)

稳定、高效工作,5V转60V的PWM电路的使用为电机提供了高效、精准的驱动信号;可将5V供电改为外接5V太阳能板,在增强设备续航能力的同时还可以节约电能,未来进一步升级改造后甚至可用于智慧农业塑料大棚的温湿度控制,具有一定的推广价值和使用价值。

## 参 考 文 献

- [1] 朱宇轩,陈丽琼.基于STM32的空气加湿器系统设计[J].无线互联科技,2022,19(3):56-57.
- [2] 段晓丽,张家超,贾支元,等.基于单片机的智能加湿器设计[J].山西电子技术,2022(4):46-48.
- [3] 叶步云,王保银.通信直流供电系统蓄电池充放电电路改造[J].现代工业经济和信息化,2021,11(8):43-44.

作者简介:闫治宇(1989—),男,硕士,讲师,研究方向为自动控制、模式识别等。

黄一哲(1994—),男,硕士,助教,研究方向为自动控制、图像处理等。

- [4] 陈宁锴.超声雾化器ASIC中核心电路的设计[D].成都:西南交通大学,2021.
- [5] 付文新,王洪丰.基于STM32单片机和DHT11温湿度传感器的温湿度采集系统的设计与实现[J].光源与照明,2022(3):119-121.
- [6] 丁慧敏.OLED显示屏驱动芯片视效优化设计与研究[D].西安:西安电子科技大学,2021.
- [7] 王波,王晋美,孙毅浩,等.基于STM32单片机的自动浇花系统设计[J].现代信息科技,2023,7(12):166-169.
- [8] 宋兴鑫,宋增凤,王秀鑫.浅谈MOS管的开关速度与功耗、EMI的关系[J].内燃机与配件,2023(2):103-105.
- [9] 王莉,陆安江.基于AT89C51的智能加湿器的设计与仿真[J].电子设计工程,2023,31(4):169-173.
- [10] 娄舒展.基于过程控制方法的智能加湿器设计研究[D].沈阳:沈阳理工大学,2023.