

四足机器人校园巡逻自适应视觉方法研究

张萌迪¹, 黄姝娟^{1,a,*}, 吕熙东¹, 柯润泽¹,

¹ 西安工业大学, 陕西, 西安, 中国

^a 349242386@qq.com

摘要: 随着人工智能专业以及机器人技术的快速发展, 人工智能专业创新创业实训将四足机器人校园巡逻方法纳入到课程教学体系当中。四足机器人凭借优异的地形适应能力与运动灵活性, 成为校园巡逻等半结构化场景巡检的重要载体。本文以四足机器人校园巡逻为应用场景, 阐述基于自适应视觉感知的巡检系统从方案论证到落地测试的全流程。本系统以 Unitree Robotics Go2 EDU 四足机器人作为研究对象, 按照视觉层与控制层分开设计的两层解耦方案搭建起“感知—决策—执行”的完整软硬结合的层次式整体框架, 实现了基于 HSV 颜色空间的自适应寻线、棕色标志点指令化变向以及红色台阶区域爬阶模式转换, 以及绿色障碍物视觉避障等功能。实验结果表明, 在典型的校园仿真场景下本系统的各个功能模块识别准确率达到 80% 以上, 全流程的任务完成率超过 90%。实现了算法研究-电路开发-程序编写-实验测试的一整套流程, 并锻炼了多学科交叉合作、软硬件联合设计以及系统的反复修改完善的能力, 在校企协同育人模式下的机器人专业教学中可以作为范例。

关键词: 具身智能; HSV 色彩空间; 感知决策执行架构; 自适应视觉感知; 四足机器人

基金信息: 基金项目: 西安工业大学陕西省省级大学生创新计划训练项目 (No.S202510702108), 西安工业大学教学改革研究项目 (No.25JBGS07)

作者简介: 张萌迪 (2003-), 男, 本科, 研究方向为人工智能; 通讯作者, 黄姝娟 (1975-), 女, 副教授, 研究方向为嵌入式计算与人工智能。

Research on Adaptive Visual Method for Quadruped Robot Campus Patrol

Mengdi Zhang¹, Shujuan Huang^{1,a,*}, Xidong Lv¹, Runze Ke¹

¹Xi'an Technological University, Xi'an, Shaanxi, China

^a 349242386@qq.com

Abstract: With the rapid development of artificial intelligence majors and robotics technology, the innovation and entrepreneurship training of artificial intelligence majors has incorporated the quadruped robot campus patrol method into the curriculum teaching system. With excellent terrain adaptability and movement flexibility, quadruped robots have become an important carrier for patrolling semi-structured scenes such as campus patrols. This article takes the four-legged robot campus patrol application scenario to explain the whole process of the inspection system based on adaptive visual perception from scheme demonstration to landing test. This system takes the

UnitreeRobotics Go2 EDU quadruped robot as the research object, and builds a hierarchical overall framework of "perception, decision-making, execution" complete soft and hard combination according to the two-layer decoupling scheme designed separately from the visual layer and the control layer, and realizes the HSV-based Adaptive line search of color space, direction change of brown sign point instruction, step mode conversion in red step area, and visual obstacle avoidance of green obstacles and other functions. The experimental results show that under the typical campus simulation scenario, the identification accuracy of each functional module of the system reaches more than 80%, and the task completion rate of the whole process exceeds 90%. It has realized a whole set of processes of algorithm research-circuit development-program writing-experimental testing, and exercised the ability of multidisciplinary cross-cooperation, joint design of software and hardware, and repeated modification and improvement of the system. It can be used as an example in the professional teaching of robots in the mode of collaborative education between schools and enterprises.

Keywords: Embodied Intelligence; HSV Color Space; Perception-Decision-Execution Architecture; Adaptive Visual Perception; Quadruped Robot

1 引言

近年来随着机器人的快速发展,具有优异地形适应性及运动灵活性优势的四足机器人被广泛应用于学校巡逻等半结构化环境下的巡检领域[1-3],而校园安全智能化的提升又催生了越来越多的自主巡检需求。目前高校工科大力推动产教融合,为打破课堂教学与产业发展之间的屏障,基于四足机器人平台进行软硬件一体化实践教学是工程类应用人才培养的有效手段之一。

然而当前同类型实训项目以及具体实施方案还存在着一定的局限性:目前常见的四足机器人路径追踪技术是基于 RGB 颜色空间阈值分割法,受环境光源影响较大,例如在校园内容易受到地面反光或者阴暗角落的影响出现误判现象,并且缺乏不同巡逻主要功能之间的转换关系。严重制约了系统的高效运转以及航行的稳定性能[4]。而且大部分实验课程设置简单、流程过短,没有涉及软件及硬件配合调试、多个项目同步操作等企业实际所需的操作培训,无法满足企业的岗位需求。

政策层面,《“十四五”数字经济发展规划》明确提出推动智能机器人在安防等领域的场景化应用[5],校园安防智能化已成为教育行业发展趋势。在此背景下,本文依托 Unitree Go2 EDU 四足机器人平台,结合校园

真实巡逻场景,设计并实现一套自适应巡线实训系统。本文完整复盘项目全流程,分析技术难点与实训实施经验,探索产教融合背景下智能机器人实训项目的建设思路,为同类实践教学项目提供参考依据。

2 方案遴选与前期准备

2.1 技术路线对比分析

轮式巡检机器人方案:开发流程简单、上手门槛低,硬件成本低且续航表现良好,但地形通过能力薄弱,无法翻越台阶、斜坡等场地障碍,适用场景单一,能实现的拓展功能有限。

RGB 视觉四足巡检方案:依托四足机器人的地形优势,场地适配范围更广;但 RGB 图像色彩、亮度信息相互绑定,光线变化、杂物遮挡极易造成识别出错,多任务同步运行时动作调度混乱,整体稳定性较差。且该方案涉及知识模块多,软硬件联调工作量大。

基于 HSV 颜色空间的四足自适应巡线方案:融合机器视觉、UDP 网络通信、多任务调度相关技术,贴合巡检机器人主流开发思路;利用 HSV 独立分离色度、亮度的特性,大幅降低光照对识别的影响,可同步集成巡线、爬台阶、避障多项功能。该方案知识点适中,调试难度适中,更加适合人工智能创新创业实训项目的开展。

综合对比各项指标, 本文最终选用基于 HSV 颜色空间的四足机器人自适应巡线方案开展系统开发。

2.2 前期学习与软硬件配置准备

在项目的开发前进行理论知识的学习, 了解 OpenCV 中对图像的一些基本处理, 如 `imread` 函数加载图片, `threshold` 阈值分割、`erode` 腐蚀、`dilate` 膨胀、`gaussianBlur` 高斯模糊以及 `findcontours` 检测轮廓等; 分析 RGB 和 HSV 的颜色空间下的图像效果, 并了解 HSV 对光照不敏感的原因; 此外, 还需要对四足机器人的层次模型中的 DDS 数据通讯方式及 SDK 使用步骤有一定认识。总结巡线、爬台阶、避障三个功能完成逻辑。查找四足机器人的视觉巡线、多任务调度的相关文献资料, 发现同类项目的共性问题, 并对以后系统的优化提供了思路: RGB 视觉容易受到光照的影响及阴影干扰, 识别不稳定; 多个模块同时进行工作会存在命令抢占导致的动作混乱等问题。

硬件主体选用宇树 Go2 EDU 四足机器人, 搭载原厂拓展坞完成外设拓展; 机器人自带 120° 广角相机、IMU 姿态传感器、足端力传感器, 内置大容量锂电池; 拓展坞集成多路 USB、串口、GPIO 接口, 可外接各类外设, 满足视觉采集、通信调试全部硬件需求。

PC 操作系统采用 Ubuntu 22.04, 安装 gcc、cmake 编译器, OpenCV 及 NumPy 视觉计算模块, ROS2 Humble 及 CycloneDDS 通讯模块; 从 GitHub 上下载 `unitree_sdk2_python` 开发库, 编译通过, 并配置好环境参数后, 设置机器人的局域网 IP 地址不变, 使用 `ping` 命令检测各台机器是否能正常通讯; 禁用遥控器对机器人的控制权, 改为让程序来控制机器人。

Python 运行环境版本不低于 3.8, 配置对应视觉、通信依赖包, 完成 SDK 示例程序调试, 实现机器人行走、步态切换、传感器读取等基础功能验证。

经过前期配置调试与理论梳理, 将整套系统拆分为图像采集与预处理、视觉目标识别、UDP 网络传输、运动指令解析、多任务调度五大模块, 明确各模块工作内容与数据

交互逻辑, 搭建完整开发框架, 降低后续整体调试难度。

3 问题分析与处理

本项目属于机器视觉与机器人运动控制相结合的综合实训内容, 实施过程中依次遇到视觉识别、网络通信、多任务调度、算力适配四大类典型问题, 结合理论知识与文献成果逐一开展调试与优化。

3.1 光照干扰问题及优化处理

项目初期沿用传统 RGB 色彩空间开展目标识别, 在自然光、灯光反光、地面污渍、树木遮挡等校园场景下, 图像像素大幅偏移, 黄色巡线、红色台阶、绿色障碍物均出现频繁漏检、误检问题, 直接导致机器人路线偏移、动作误触发。

RGB 颜色空间将色彩与亮度信息相互耦合, 对户外光照变化适应性差[6], 这也是移动机器人视觉领域的共性难题。针对光照干扰问题, 统一采用 HSV 色彩空间完成目标分割。针对黄色巡线、棕色转向标识、红色台阶、绿色障碍物分别配置专属 HSV 阈值区间。在此基础上, 对二值化图像依次执行高斯模糊、腐蚀、膨胀形态学操作完成图像降噪[7], 同时增设轮廓面积、像素坐标双重判定规则, 过滤无效干扰区域。优化后视觉模块在轻微光照变化、局部遮挡场景下均可稳定运行。

3.2 多任务指令冲突及调度优化

系统包含自适应巡线、指令转向、爬台阶、视觉避障四大功能模块, 各模块独立运行正常, 联调阶段出现指令相互抢占、步态频繁切换、动作错乱等问题。

为解决该问题, 建立多级任务优先级管控体系[8], 优先级从高至低依次为: 爬台阶 > 避障 > 指令转向 > 常规巡线。高优先级指令触发后自动屏蔽低优先级逻辑; 同时为爬台阶等功能增设延时防误触发、状态标记机制, 避免短暂视觉丢失造成步态误切换, 实现多模块有序协同运行。

4 系统实现

4.1 系统总体设计

该系统是采用分层次设计思想而建立起来的一个整体框架，整个系统由感知层、决策层、执行层三层构成如图 1 所示。

感知层是整个系统的最基础也是最重要的信息来源部分，使用以视觉为主的感觉方式来保证对环境感知的细致性和准确性：机器人头部安装普通的相机摄像头获取场景中的图像纹理信息用于描述环境；机器人体内有高精度的惯性测量单元(IMU)以及轮式里程计用来获得机器人的各个方向的角度、线加速度还有位置的变化以便进行下一步的动作规划。

决策层部分使用高性能嵌入式芯片(NVIDIA Jetson Xavier NX)，负责图像处理、

视觉信息分析以及运动策略生成等工作。将整个系统划分为两层，即视觉层和控制层，在视觉层中利用 Python 编写相机抓取、目标检测等相关程序，在控制层中通过 C++调用底层 SDK 完成对于机器人的运动闭环控制，二者之间使用 UDP 进行交互。这极大地降低了各个模块之间的耦合性，提高了系统的可扩展性和稳定性。而这种感知与控制分离的思想也是足式机器人在复杂地面行走的一种有效方法[9]，对于本课题组后续开展自主巡检具有十分重要的意义。其系统框图如图 2 所示。

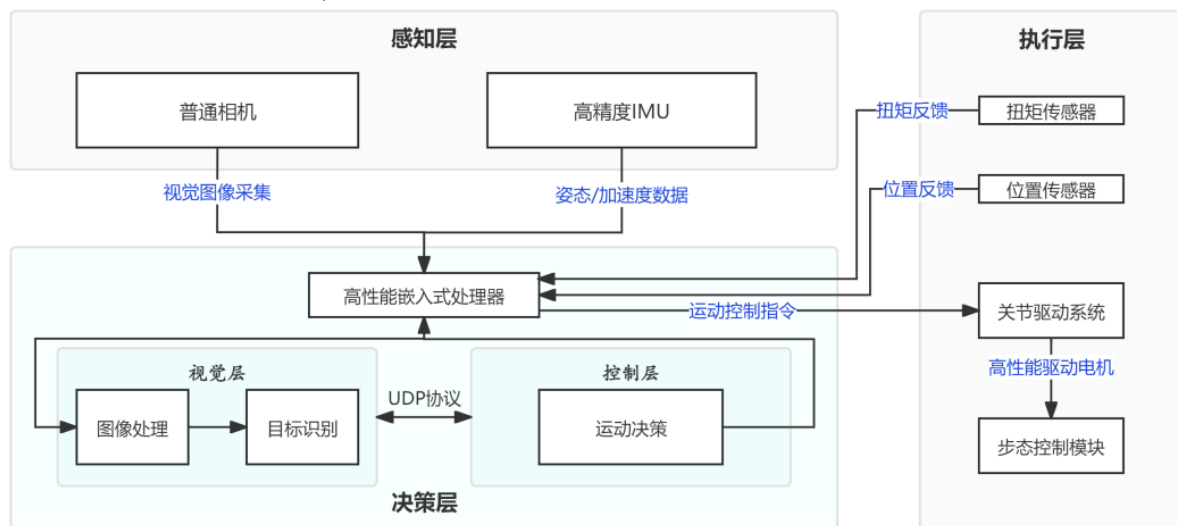


图1. 系统分层设计架构图

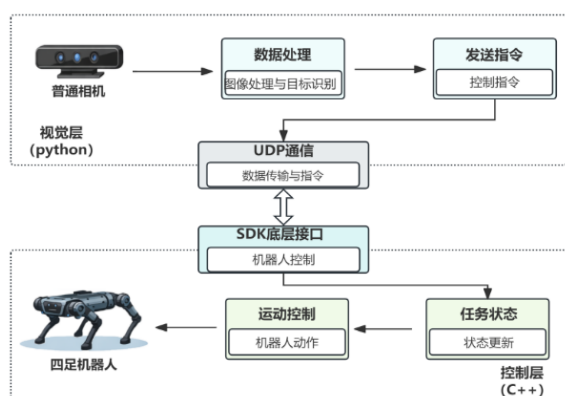


图2. 决策层内部架构图

执行层由四足机器人的关节驱动器及步态控制器组成。该机器人使用 12 个大功率驱动电机，每个关节的最大驱动力矩达到 $45\text{N}\cdot\text{m}$ ，在各个关节的角度被准确地控制下，可以实现身体($-48^\circ\sim 48^\circ$)、大腿($-200^\circ\sim 90^\circ$)、小腿($-156^\circ\sim -48^\circ$)的大范围摆动，从而提高对

于各种地面的适应性和执行各种复杂的任务的能力。并且根据上层决策的结果做出相应的动作；同时在每个腿关节处装有扭矩传感器和位置传感器用于反馈关节当前的状态并据此来进行步态调整以及力的控制。

感知层与执行层为系统运作提供最基本的感知信息以及动作反馈，而在它们之间起着纽带作用的是决策层，它承担着诸如核心算法计算等大量工作，这部分的设计会对系统自主巡逻的效果及鲁棒性产生影响。因此，在本文中将对决策层进行更深入探讨并对其相关功能模块进行介绍。

4.2 核心代码函数说明

本系统决策层的算法实现以及运动控制都基于开源库和硬件 SDK 接口编写，主要使用的函数及其作用见表 1，以后各部分的设计都将使用这些函数。

表 1. 系统核心代码函数及功能描述

函数名称	主要用途
socket	创建 UDP socket 用于网络通信
bind	绑定本地端口
recvfrom	接收 UDP 数据包
sendto	发送 UDP 数据
setblocking	设置 socket 阻塞模式
Thread	创建线程
sleep	线程延时控制
np.array	创建数组 (设定 HSV 阈值)
cv2.cvtColor	颜色空间转换 (BGR 转 HSV)
cv2.inRange	基于 HSV 阈值创建颜色掩码
cv2.erode	图像腐蚀操作
cv2.dilate	图像膨胀操作
cv2.GaussianBlur	高斯模糊降噪
cv2.findContours	查找图像轮廓
cv2.moments	计算轮廓矩 (求质心)
ChannelFactoryInitialize	初始化通信通道
SportClient	创建运动客户端
Init	初始化客户端
GetImageSample	获取图像样本
Move	控制机器人移动
StopMove	停止机器人移动
SwitchGait	切换步态模式
optimization_option	图像优化处理 (形态学+高斯模糊)
color_option	多颜色阈值分割
detect_brown_marker	棕色右转标识检测
GetInitState	获取机器人初始状态
VelocityMove	速度控制移动 (封装)
ClimbingStair	控制爬楼梯行为
StopClimbing	停止爬楼梯行为
ObstaclesAvoid	执行避障绕行序列

4.3 自适应巡线与指令化转向模块算法设计

为了使宇树(Unitree) GO2 四足机器人能够按照指定路线自动行走,采用普通的摄像头作为主要传感器,根据其拍摄的实时视频信息设计了一套基于颜色追踪方法的自学习寻迹控制策略,具有编程简单、设备价格低廉以及延迟时间短的优点,适用于定线环境中的自主寻路;主要采用图像处理技术和运动控制技术进行反馈循环控制以保证车辆沿着黄色轨迹线完成自动寻迹。

图像预处理部分将摄像头捕获的 RGB 图像转换为 HSV 颜色空间以消除由于光照

的变化而对 RGB 空间的敏感性。对于由光照变化引起的不利因素,在图像预处理阶段进行颜色空间的优化是提高识别可靠性的有效途径之一[10]。

故分别设置黄色赛道线与棕色直角转弯标识精确色阈(黄 H: 16~50 S: 60~255 V: 100~255 棕 H: 5~25 S: 80~255 V: 50~200),利用 cv2.InRange 对两颜色域进行阈值处理并采用机器视觉中常用图像去噪技术形态学运算(腐蚀—膨胀)和高斯滤波(5*5 卷积核)去噪二值图,有效地抑制了复杂光照情况下的噪声干扰,增强黄色赛道边缘以及棕色标志的提取精度。自适应巡线流程图如图 3 所示。其代码如图 4 所示。

在决策层面上,算法首先确定分辨率为 640×480 的图像的有效检测范围(纵坐标 280~478,横坐标 150~635),首先判断是否有棕色直角右转标志:将棕色掩膜图进行轮廓识别(cv2.findContours),获取其轮廓面积;若存在有效棕色轮廓,则为右转结点,向右直角转弯-0.8rad/s,控制机器人狗做直角右转操作,直角转弯后回到正常巡线逻辑。如果没有找到棕色直角右转标志牌,则进入正常的黄色巡线逻辑:使用轮廓检测得到黄色部分的质心(xc,yc),其中质心公式为公式(1)[11]:

$$\begin{cases} x_c = \frac{M_{10}}{M_{00}} \\ y_c = \frac{M_{01}}{M_{00}} \end{cases} \quad (1)$$

式中 M_{00} 为零阶矩,等于二值图区域内像素总数, M_{10} 为水平一阶矩, M_{01} 为竖直一阶矩。视觉决策层与运动执行层间指令传递采用 UDP(4002/4003)协议完成:视觉部分实时求解误差并将运动命令发送给运动部分,而运动控制部分则根据 unitree_sdk2py 中的 SportClient 接收并解读运动命令。根据检测到的图像引导机器人运动,实现“获取图像 - 提取信息 - 计算偏差 - 运动执行”,并设置命令响应超时判断,当超过设定时间仍未收到其他运动控制命令,则开始巡线运动以保证其安全性及稳定性(设定时间为 2s)。

该算法通过质心偏差的实时反馈替代传统 PID 控制的复杂参数整定,以轻量化的偏差决策逻辑实现路径跟踪,既降低了计算资源消耗,又保障了在 0.3 m/s 巡航速度下的路径跟踪精度。

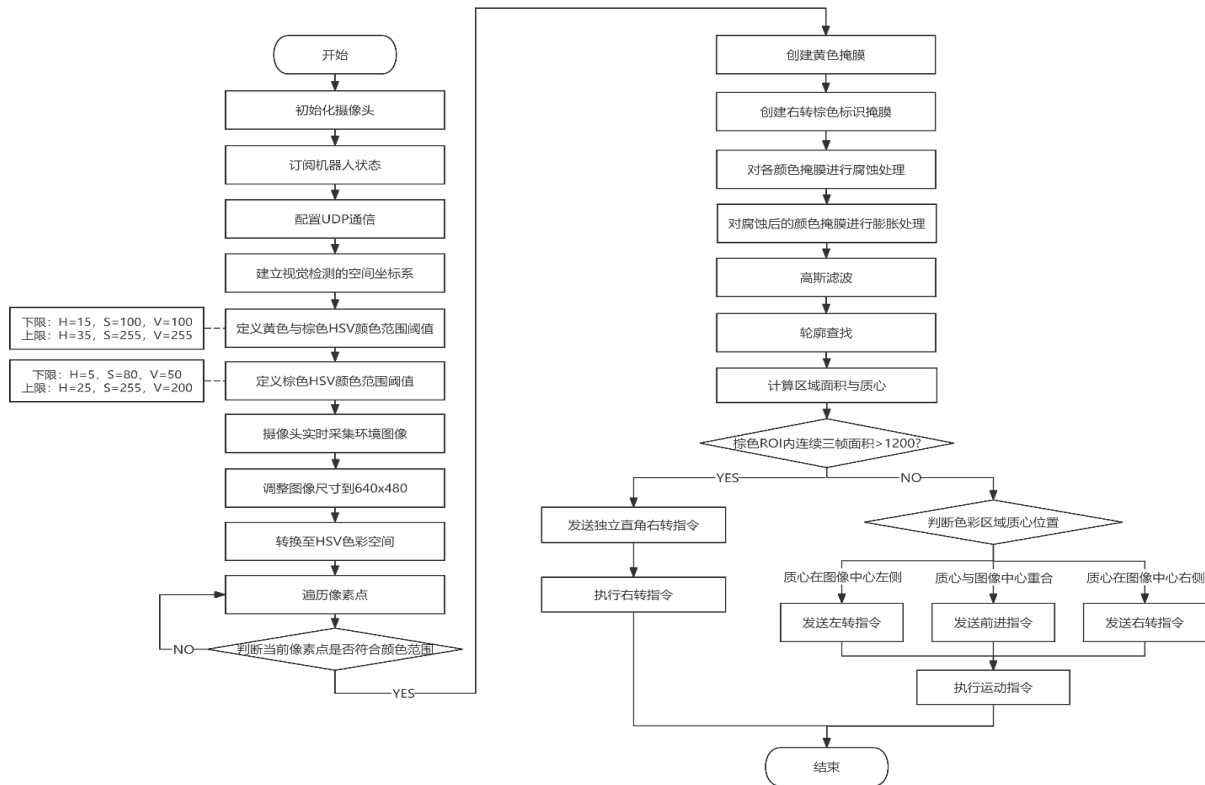


图3. 自适应巡线模块流程图

```

# 图像预处理
def optimization_option(binary_img):
    kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (9, 9))
    eroded = cv2.erode(binary_img, kernel)
    dilated = cv2.dilate(eroded, kernel)
    blurred = cv2.GaussianBlur(dilated, (5, 5), 1)
    return blurred

def color_option(color_image):
    hsv = cv2.cvtColor(color_image, cv2.COLOR_BGR2HSV)

    # 棕色区域 (卸货)
    lower_pink = np.array([165, 40, 180])
    upper_pink = np.array([178, 140, 255])

    # 黄色区域
    lower_yellow = np.array([16, 60, 180])
    upper_yellow = np.array([50, 255, 255])

    # 绿色障碍
    lower_green = np.array([35, 50, 50])
    upper_green = np.array([90, 255, 255])

```

图4. 图像预处理核心代码片段

4.4 爬台阶模块设计

教室走廊中有台阶等高低落差，在经过台阶这一步骤是完成整个校园巡视工作的重要一环。RGB 视觉法采用预先处理加后期处理的方式，在机器人移动过程中可以检测到台阶并能顺利通过[12]。所以本文仿真环境中，把楼梯区域着色成红色，使其与其他部分有明显区别便于识别区域。

本模块通过颜色对比度以及设定红色区域面积阈值实现对台阶区域的快速检测，在此基础上添加延迟以及优先级控制等，保证机器人上楼梯平稳性和连贯性。其工作过程

如图 5 所示。

4.4.1 台阶区域视觉识别逻辑

系统利用对图像中红色部分进行观察来检测台阶，主要过程为：

(1) 颜色阈值配置：红色 HSV 阈值

分为两个区间：第一个区间下限[0, 70, 60]到上限[10, 255, 255]，第二个区间下限[170, 70, 60]到上限[180, 255, 255]以便更好地识别出台阶区域并且与其他巡线（黄色）或避障（绿色）等场合的颜色阈值相区分。

(2) 图像预处理与轮廓筛选

对红色掩膜图像依次进行腐蚀、膨胀、高斯模糊等形态学处理，在去除环境噪声的基础上获取轮廓；只当该轮廓面积大于 30000 才认为是台阶，以排除地面上一些较小红色区域的影响，代码如下图 6 所示。

(3) 3 秒延时防误触发机制

如果检测到有效的红色块，则“爬楼 $stop_time = current_time + 3s$ ”，并且发送爬楼指令；如果红色块暂时未被检测但 $time$ 还未达到 3s，则持续发送爬楼指令，在 $time$ 到达 3s 后停止运动以防因楼梯缝隙等造成步态变化。

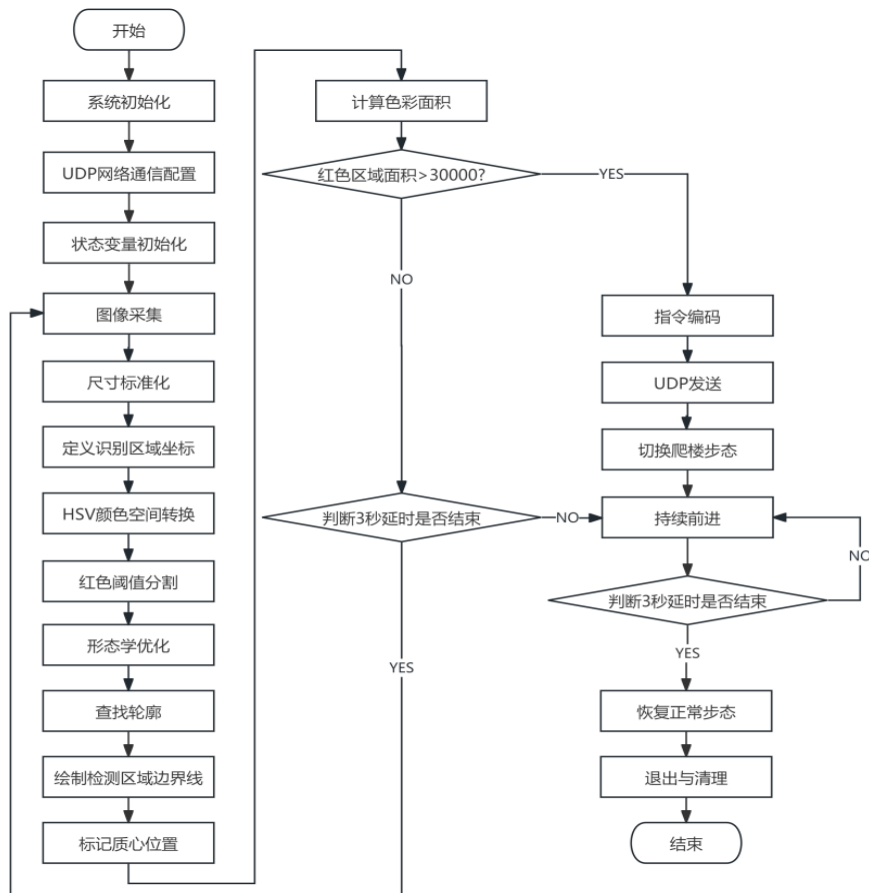


图5. 爬台阶模块完整执行流程图

4.4.2 爬楼运动控制逻辑

运动控制器将爬楼作为最高优先级的任务，使主要的动作不会受到其他任务的影响：

(1) 步态切换与持续运动：

复杂的路面条件下步态转换平滑度对于机器人的顺利跨越障碍至关重要，在此采用基于相位的步态转换方法[13]，在第一次接收到上楼信号后立即转换为上楼专用步态（Gait 3）并设置为上楼状态，同时发送 0.2 m/s 前进速度命令，以使机器人有足够的动力进行上楼操作并且利用步态过渡减小机身姿态变化，其程序示意图如图 7 所示。

(2) 状态复位机制

当爬楼指令结束并且之前系统是在爬楼状态下，则立刻停止行走，回到正常步态（Gait 1），同时清空爬楼标志。

(3) 优先级管控

爬楼指令优先级最高，高于巡线、避障等各种动作命令，在爬楼命令不成立的情况下，才会进行相关动作。

```

# 红色检测
# =====
found_red = False
for c in cnts_red:
    if cv2.contourArea(c) > 30000:
        found_red = True
        break

if (not stair_mode_active) and found_red:
    stair_mode_active = True
    stair_mode_until = now + STAIR_MODE_DURATION
    print("Main: >>> 检测到红色区域，进入10秒爬楼独占模式")

if stair_mode_active and now == stair_mode_until:
    stair_mode_active = False
    print("Main: >>> 10秒爬楼独占模式结束，恢复正常识别")

# 基占期间：只发10、屏蔽其他所有动作
if stair_mode_active:
    spot1 = 10
    cv2.putText(
        color_image,
        f"STAIR MODE (max{0.0, stair_mode_until - now}:1f)s",
        (20, 160),
        cv2.FONT_HERSHEY_SIMPLEX,
        0.8,
        (0, 0, 255),
        2,
    )

```

图6. 图像预处理与轮廓筛选详细代码

```

def ClimbingStair(self):
    now = time.time()

    # 第一次进入爬楼模式，切步态并启动计时
    if not self.climb_mode_active:
        print("Sport: >>> 进入爬楼模式，准备切换爬楼梯步态 (Gait 3)")
        self.client.StopMove()
        time.sleep(0.8)

        self.client.SwitchGait(3)
        print("Sport: >>> 已发送第1次 SwitchGait(3)")
        time.sleep(0.8)

        self.client.SwitchGait(3)
        print("Sport: >>> 已发送第2次 SwitchGait(3)")
        time.sleep(0.8)

        self.is_climbing = True
        self.climb_mode_active = True
        self.climb_mode_end_time = time.time() + self.climb_duration
        print(f"Sport: >>> 爬楼梯步态切换完成，开始自动前进 (self.climb_duration) 秒")

```

图7. 步态切换详细代码

4.5 避障设计

避障模块采用的是纯视觉感知的检测及决策方法，这是移动机器人避障检测的一种主流实现方法[14]。其中视觉感知层在 HSV 颜色空间中运用了颜色阈值分割算法来对绿色的障碍物区域进行像素级分割，在结合轮廓面积以及像素坐标的位置阈值后可完成障碍物的初步定位并对其有效性进行检验。将“轮廓面积> 5000 像素&像素纵坐标> 300”设定为有效障碍物的判定条件，在排除了小面积杂色区域误触发的同时，也保证了只对机器人的前近方障碍物做出响应，提高了避障决策的准确性和可靠性。

根据图像处理获取到的障碍物坐标，在决策层设计局部避障动态轨迹规划算法，并结合静态与动态轨迹规划方法以及多地形移动机器人鲁棒性跟踪策略[15-16]，给出一系列操作指令：回退→左前方横向避障→直线穿越障碍区域→右前方横向返回主路线。并且通过运动控制接口来控制机器人的实际移动速度以及移动的方向(vx/vy/vyaw 多维速度闭环控制)，避障完全实现的流程图见图 8，核心程序见图 9。

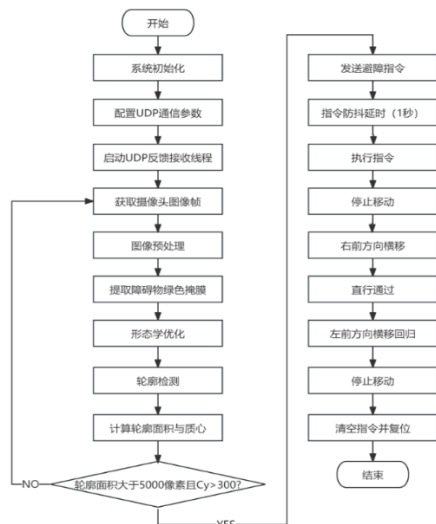


图8. 避障模块完整流程图

```
def ObstaclesAvoid(self):
    print("Sport: >>> 收到避障指令，开始执行避障序列...")
    self.client.StopMove()
    time.sleep(0.5)
    for _ in range(200):
        self.client.Move(0.2, -0.4, 0)
        time.sleep(0.01)
    for _ in range(300):
        self.client.Move(0.3, 0, 0)
        time.sleep(0.01)
    for _ in range(200):
        self.client.Move(0.2, 0.4, 0)
        time.sleep(0.01)
    self.client.StopMove()
    print("Sport: >>> 避障动作结束")
```

图9. 避障模块核心代码

5 测试分析与实训总结反思

5.1 测试环境与标定工作

为了检验系统的性能，本文设计了一个长为 4500mm，宽为 3500mm 的测试场地，并将黄色的线条设置为巡逻路径，实现起点—便利店—实验楼—图书馆—篮球场—终点这一巡逻任务；同时，为了体现现实情况中复杂性和多变性的特点，在该场地设置了台阶区、障碍区等功能区，可以用于四足机器人的自适应步态切换来应对复杂的地形环境，整体路径规划如图 10 所示。

为了保证机器人的跟踪效果，本文选取了 10cm 宽的常规 PVC 警示胶带作为黄色巡线材料，基于颜色特征的巡线方式可以保证四足机器人经过全部的巡检区域。搭载普通摄像头，四足机器人可以采集当前场景下的图像数据，并进行颜色变换及图像处理。可以对复杂环境中的指定颜色进行路径追踪及标志识别[17]。本文实验采用仿真实验室，在巡线、爬台阶以及避障这三块区域分别进行了多次实验。

实验设置正常光照、轻微反光及局部遮挡等不同环境条件，每种场景均重复测试 30 次，以保证实验结果的统计有效性。

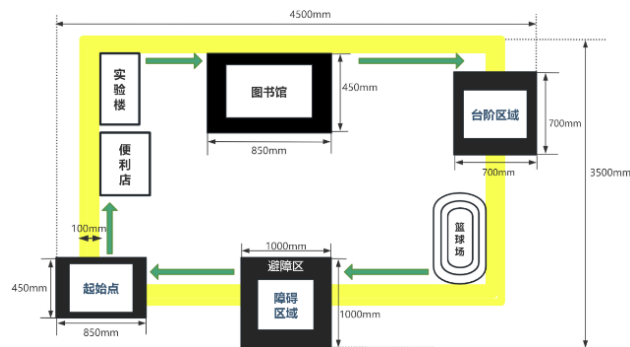


图10. 校园场地模拟环境

5.2 实验评价指标

为客观评价各模块性能，设定识别准确率、动作执行成功率、单次任务完成耗时、连续运行稳定性四大核心评价指标，其中识别准确率为目标区域/物体正确识别次数与总测试次数的比值，动作执行成功率为模块功能完整实现次数与总测试次数的比值，连续运行稳定性以机器人无故障连续完成 10 次全流程任务为合格标准，各模块测试均重复进行 30 次，取平均值作为最终结果。

5.3 自动巡线模块实验结果与分析

自动巡线模块以黄色赛道为识别目标，需实现机器人在无遮挡、轻微光照变化环境

下的精准路径跟踪，同时避开灰色、绿色、棕色等障碍线。实验设置正常光照、轻微反光、局部遮挡 3 种测试场景，各场景测试 10 次，实验结果如表 2 所示。

表 2. 自动巡线模块实验结果

测试场景	识别准确率 (%)	动作执行成功率 (%)	平均巡线速度 (m/s)	单次巡线平均耗时 (s)
正常光照	85	99	0.3	25.2
轻微反光	80	90	0.29	26.7
局部遮挡 (障碍线旁)	83	95	0.28	28.1

实验结果表明，在正常光照无遮挡场景下，基于 HSV 色彩空间的色阈检测算法可实现黄色赛道的 85% 精准识别。机器人通过统计目标区域黄色像素分布调整行进方向，直行、左转、右转动作衔接流畅，以 0.3m/s 的速度沿赛道稳定巡线，无偏移、卡顿现象，满足路径跟踪需求，本模块实物状况如图 11 所示。轻度反光场景：由于赛道区域内部分像素被反光偏移，识别正确率降低到 85%，机器人出现 2 次小幅度方向误差；可以通过设置 HSV 的阈值上下限将识别正确率提高到 83% 以上。障碍线局部遮挡场景：通过轮廓检测及面积阈值过滤障碍线干扰，该算法的识别正确率仍能保持 83%。

5.4 爬台阶模块实验结果与分析

爬台阶模块以红色色块为台阶区域识别标识，需实现机器人从平地步态向爬楼步态的精准切换，完成 60mm 高台阶的稳定跨越后恢复常规步态。实验设置无干扰、轻微红色干扰 2 种测试场景，各场景测试 10 次，实验结果如表 3 所示。



图11. 机器狗的自动巡线模式

表 3. 爬台阶模块实验结果

测试场景	识别准确率 (%)	动作执行成功率 (%)	爬楼平均速度 (m/s)	单次爬台阶平均耗时 (s)
无干扰	88	100	0.2	15.4
手机手电筒干扰	83	100	0.2	15.4

实验结果显示，在无干扰场景下，算法通过设定红色区域面积阈值 (30000 像素) 实现台阶区域的 88% 精准识别，机器人接收到爬楼指令后，调用 SwitchGait(3) 函数快速切换至正向爬楼步态，以 0.2m/s 的速度平稳跨越台阶，四肢步态协调，无打滑、倾倒现象；通过台阶后红色区域消失 10 秒，机器人自动恢复常规步态，动作执行成功率 98%，单次爬台阶耗时稳定在 15.4s 左右。

在手机手电筒干扰场景下，算法通过面积阈值过滤小面积干扰色块，识别准确率保持 83%，无爬楼动作失败情况。爬台阶模式下的机器狗如图 12 所示。



图12. 机器狗的爬台阶模式

5.5 避障模块实验结果与分析

避障模块采用视觉识别的检测方式，以绿色为障碍物视觉标识，实时采集环境图像信息，机器人需通过组合运动动作完成障碍物绕行并回归原黄色巡线赛道。实验设置无遮挡、轻微遮挡 2 种测试场景，各场景测试 10 次，实验结果如表 4 所示。

表 4. 避障模块实验结果

测试场景	识别准确率 (%)	动作执行成功率 (%)	避障平均速度 (m/s)	单次避障平均耗时 (s)
无遮挡	87	98	0.23	8.2
轻微遮挡 (障碍物局部遮挡)	85	95	0.23	8.2

在无遮挡场景下，视觉识别检测能够实现障碍物的 87% 精准识别。机器人接收到避障指令后，执行“停止→右前横移→直行通过→左前回归→停止”的组合动作序列，动作衔接流畅，无碰撞、偏移现象，成功绕开 300mm 左右规格障碍物并回归黄色赛道；避障完成后巡线逻辑无缝接管，动作执行成功率 100%，单次避障平均耗时 8.2s。

在轻微遮挡场景下，障碍物被局部遮挡导致视觉识别的绿色区域面积减小，算法通过面积阈值(5000 像素)与位置阈值($cY > 420$)的双重验证，过滤虚假识别信息，识别准确率保持 85%；机器人避障速度保持为 0.23m/s，确保了绕行过程的稳定性。

实验表明，视觉识别检测方式能够提升障碍物检测的精准性与可靠性；预设的组合运动动作序列可保证机器人在障碍区域的稳定绕行，避障完成后清除指令并重置超时保护，确保巡线逻辑的正常接管，实现避障模块与巡线模块的无缝衔接。避障模式下的机器狗如图 13 所示。

待上述各个子系统的功能验证及参数调试完成后，进行整体串联调试，指令机器人从起点处开始，“沿路巡检(经过篮球场、图书馆、实验楼等)→上台阶→绕过障碍物→返回原点”，并连续重复该过程共 10 次。结果表明：10 次运行过程中，机器人的整个流程任务完成度达到了 90%，共完成了 9 次无故障的任务全部完成，其平均总时间为 100.0s。



图13. 机器狗的避障模式

全流程测试中，各模块的衔接响应时间均控制在 5.0s 以内，视觉层与控制层通过 UDP 协议实现数据的实时交互，无指令延迟、丢失现象；机器人在各功能区域的动作执行精准，爬台阶、避障等动作均按预设流程完成，无姿态不稳情况。

5.6 整机综合测试

本次实验结合四足机器人仿真领域的相关研究方法[18]，在完成单模块测试后开展 10 轮全流程整机测试，模拟完整校园巡逻任务。测试结果显示，系统整体任务完成率为 90%，各模块切换响应时间均小于 5.0 s，UDP 通信无丢包、延迟问题，整机各项指标达到设计要求。

6 总结与展望

6.1 结论

本课题作为校企合作开发的人工智能专业创新创业实践课程的一部分，选用 Unitree Go2 EDU 四足机器人进行实验测试，采用 HSV 颜色空间算法结合阈值分割法完成路径跟踪功能，并根据传统四足机器人巡线方法中使用轮式巡检机器人和 RGB 视觉两种方案的特点，在实际应用过程中针对环境光源变化对图像采集的影响、UDP 协议传输过程中的丢包、多个控制命令同时下发导致执行混乱的问题进行分析解决。将上述三种方法(基于 HSV 颜色空间分割法、通信协议优化设计以及三级任务优先级调度策略)分别应用到机器人中，实现具有自动巡线、爬楼梯及自动避障功能的自主巡逻机器人。

搭建模拟实验场地开展多工况对照测试，分模块完成不同光照、遮挡条件下性能验证，再实施整机全流程联调。实验数据表明，系统各功能模块识别稳定、动作执行流畅，整机完整巡逻任务完成率可达 90%，模块切换响应及时，通信传输稳定，能够为学生提供

实训项目研究内容。

整套开发流程完整还原智能巡检机器人从方案论证、程序开发、软硬件联调到整机性能测试的完整工程流程,融合数字图像处理、网络通信、机器人运动控制等多领域专业知识,能够完整训练工程问题分析、软硬件协同调试、故障排查与系统迭代优化等实操能力。

6.2 展望

本次实训项目采用固定 HSV 阈值实现色彩分割,仅适用于实验场地常规光照条件,在强光直射、大面积阴影等极端场景下识别精度还有待提高;这给学生留有思考和创新空间。系统仅搭载单目视觉采集图像,未搭配深度、红外辅助传感设备,为后续内容进一步拓展做准备。

本项目使得学生掌握 RGB、HSV、网络通信等理论知识后,然后进行分层模块化拆分开发。项目整体技术难度较高,同时叠加视觉、通信、运动控制多任务处理,调试工作量大,能够支撑实训项目课程。且开发过程可以参考宇树官方开源开发文档完成设计,整体功能在室内演示没有问题,在室外复杂场景下进行设计将是下一步提升学生能力的研究内容。

参考文献

- [1] 陈晓东,梁永康,连宗凯.防爆四足机器人在成品油库的自主巡检系统设计与安全评估[J].产品可靠性报告,2025,(10):37-38.
- [2] 付鑫,朱西硕.煤矿四足机器人巡检系统设计与研发[J].智能矿山.2025,(11):71-75..
- [3] 王堃,唐清琼,付环.面向 LNG 接收站的四足机器人巡检系统设计与研究[J].电器工业.2025,(8):71-75.
- [4] 贾琪,李艳,徐洋,等.基于中枢预测控制的四足机器人多地形运动控制[J].东华大学学报(自然科学版),2026,(7):1-16.
- [5] 国家互联网信息办公室.“十四五”数字经济发展规划 [Z].2022.
- [6] Sridharan M, Stone P. Color learning and illumination invariance on mobile robots: A survey[J]. Robotics and Autonomous Systems, 2009,57(6-7):629-644.
- [7] Mur artal Raúl, Montiel J.M.M, Tardós Juan D. ORB SLAM: A versatile and accurate monocular SLAM system[J]. IEEE Transactions on Robotics, 2015, 31(5): 1147-1163.
- [8] 同爱丽.实时多任务调度方法研究与应用 [D].西安:西北工业大学,2006.
- [9] Lee Joonho, Bjelonic Marko, Reske Alexander, et al. Learning robust autonomous navigation and locomotion for wheeled-legged robots [J]. Science Robotics,2024,9:eadi9641.
- [10] 孙帅帅,冯春晓,张良.基于离散采样的多模态四足机器人路径规划[J].吉林大学学报:工学版.2025,(11):3736-3744.
- [11] 章敏晋.图像工程(上册):图像获取与预处理[M].北京:清华大学出版社,2012.
- [12] Wozniak Piotr, Penar Paweł, Bielecki Damian. RGB-Based Staircase Detection for Quadrupedal Robots: Implementation and Analysis[J]. Sensors, 2025, 25(23): 7247.
- [13] 曾明海,李澍源.基于相位指导的四足机器人步态切换方法研究[J].机械设计与制造,2026,(7):286-291.
- [14] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots[C] //Proceedings of 1985 IEEE International Conference on Robotics and Automation. St. Louis: IEEE, 1985:500-505.
- [15] 王康.四足机器人复杂地形越障控制方法研究[D].北京:北京交通大学,2023.
- [16] 修春良,范永,刘大宇,等.四足机器人动静态路径规划方法研究[J].现代信息科技,2025,9(24):192-198.
- [17] 杨婉霖,贾豫东.四足机器人颜色识别算法设计[J].计算机科学与应用, 2024, 14(12):111-117.
- [18] 董夏男,蒋学程.URDF 与 Simulink 结合的四足机器人仿真研究[J].机械管理开发,2025,40(08):272-275.