工学

自主运输无人机的视觉识别系统的设计

北京物资学院 何浩源 张胜磊 李秋辰

唐恒亮 副教授

摘要：

 空中机器人是指各种由动力驱动、机上无人驾驶、可重复使用的、能够在空中自主飞行的无人飞行器。我们组本次完成的这个项目，主要以现有的开源硬件和飞控系统为基础，通过编写相应的程序，从而实现机器人的自主起飞、自主循迹、自主抓取物品、自主降落等任务。

关键词：空中机器人；四轴飞行器；飞行控制系统。

一、项目背景

 无人机也称无人航空器或遥控驾驶航空器，是一种由无线电遥控设备控制，或由预编程序操纵的非载人飞行器，拥有众多有人驾驶飞机所不具有的优点。例如体积小、成本低、适应性强、机动性好等优点，已经在抢险救灾、电力巡线、林业、以及军事等领域得到广泛应用[1]。因此，对于完成自主运输无人机的视觉识别系统的设计的这个项目，有着十分广阔的科研和应用前景[2]。

 本项目以现有的机器人比赛当中的飞行器类的题目为背景，针对自主起飞、自主循迹、自主抓取物品、自主降落等常见的任务为目标进行的研究。通过本次项目的研究，可以为将来相关的机器人比赛，打下良好的基础。同时，也可以锻炼我们的硬件设计和组装能力，相关软件的编写能力。所以，我们团队在参考了2017年华北五省（市、自治区）大学生机器人大赛当中的空中机器人比赛的题目要求，对题目进行了相应的简化，从而设计出了我们团队在本次大学生科学研究与创业行动计划项目当中的创新训练项目形式当中的总体目标。并在为期半年的时间内，完成了此次项目。

二、研究内容

（一）研究内容

 我团队组装的空中机器人主要涉及到空中飞行器、视觉传感器、飞行器控制系统和机械臂等。在完成这个大创项目的过程当中，主要涉及到了无人机的自动控制和视觉循迹这两个主要的组成部分。其实现主要通过现有的硬件和飞控系统为基础，通过将Pixhawk飞控硬件系统所运行的飞控程序，是由ArduPilot团队编写的，开源的ArduCopter飞控程序，与树莓派所搭载的摄像头和开源的OpenCV计算机视觉库程序相结合。其中，利用了Mavlink无人机通讯协议，用Python语言编写了相应的控制程序，将两者项结合在了一起，从而实现了无人机的自主巡线的功能。

（二）研究目标

1、通过视觉识别实现无人机的自主循迹功能。

2、无人机可依靠预定的程序，自主拾取物品。

3、无人机可依靠视觉识别自主到达指定地点，完成运输和投放物品。

（三）解决的关键问题

 在本次项目当中，在硬件方面，我们主要通过选购相关的硬件系统，例如Pixhawk、光流模块、F450机架套装、树莓派和电脑拆机摄像头等硬件，并以此为基础，对其进行改造，以便符合我们对于四轴无人机所需硬件的要求。在软件方面，我们主要通过使用现有的开源软件，例如ArduCopter飞控系统和基于Linux系统，运行在Python环境下的OpenCV计算机视觉库。在ArduCopter和自动飞行的控制程序之间，通过MavLink相连。

 本次项目所研究的内容的关键，主要是如何编写相应的程序，将硬件系统与软件系统相融合。从而使自动飞行的控制程序，能够实现四轴飞行器的自主循迹功能和物品的运输功能。

三、技术路线及解决方案

（一）整体可行性分析

**1、四轴飞行器的可行性分析**

 经过调查，我们发现，四轴飞行器的稳定性和可控制性都比较好，所以，我们选择了四轴飞行器，作为我们飞行器的基础平台。

（二）飞行器设计部分

**1、选择四轴飞行器的原因**

 四轴飞行器相对于其他飞行器程序控制简单，开源支持度广，生态系统完整性更强。因此，我们团队选择了四轴飞行器作为自主运输无人机的视觉识别系统的设计当中。

**2、自主运输无人机的视觉识别系统的设计的总体流程图**



（三）飞行器的硬件组成

 飞行器的硬件主要由Pixhawk、树莓派、机架、摄像头和机械臂等硬件组成。

**1、Pixhawk**

 Pixhawk是一款由PX4开源项目设计并由3DR公司制造生产的高级自动驾驶仪系统[3]。我团队选择Pixhawk作为飞控的原因，主要是其开源的特性，参考文档相对较多。并且，国产版本的Pixhawk在其价格方面，也具有一定的价格优势。

 在四轴飞行器的飞控方面，我们团队选择了Pixhawk作为飞控系统。在Pixhawk的配套硬件方面，我们团队选择了一套适配现有的Pixhawk的光流模块。此光流模块，同时具有I2C和串口两种接口。在通过I2C接口向飞控提供光流所需的数据的同时，还可以通过串口连接到树莓派。以便在自动控制程序当中，读取四轴飞行器的高度数据。

**2、树莓派**

 树莓派通常指的是Raspberry Pi电脑板，中文的译名是“树莓派”。在树莓派所搭载的系统方面，Raspbian是目前树莓派基金会所官方支持的Linux操作系统的版本。

 在四轴飞行器的自动控制的硬件方面，我们团队选择了树莓派的3B版本和树莓派的3B+版本作为四轴飞行器的自动控制的硬件平台，并在以此为基础上，增加了USB转TTL串口模块和电脑拆机的摄像头模块。USB转TTL串口模块的与光流模块相连，用于读取光流模块所提供的四轴飞行器的高度数据。电脑拆机的摄像头模块，通过USB数据线与树莓派相连，以便提供图像给OpenCV来进行处理。OpenCV将图像数据处理完成后，以便给主程序提供自动控制所需的数据。

**3、机架、摄像头和机械臂等硬件**

 由于项目的时间和经费有限，故因此选用了价格适中的F450机架套装和F450所适配的脚架，并为此选配了4个30A电调、4个V2216，900KV无刷电机、2对1045正反桨和5200mAh的3S，XT60插头的电池。

 在图像的采集方面，我们团队采用了电脑的拆机的摄像头，并用数据线与树莓派相连。通过串口，来读取摄像头采集到的图像，以便OpenCV读取并使用相关的数据。

 在机械臂的设计上，我们团队使用了铁丝弯折而成的钩子，并通过使用纸箱的纸板固定，电工胶带和铁丝相互粘结，从而固定了钩子的方向。并对钩子的形状进行了相应的设计，以便机械臂的收起和展开，从而避免了在起飞和降落的过程当中，影响四轴飞行器的起飞和降落。

（四）飞行器的软件组成

 飞行器的软件主要由ApmCopter飞控系统、MavLink协议、OpenCV计算机视觉库和其余的软件组成。

**1、ApmCopter飞控系统**

 ApmCopter是由ArduPilot团队编写的一款开源的飞控系统。因为其开源的特性，并经历了长期的开发，现已成为一款在飞控当中大规模使用的飞控程序。

**2、MavLink协议**

 Mavlink协议是在串口通讯基础上的一种更高层的开源通讯协议，主要应用在微型飞行器（micro aerial vehicle）的通讯上。

 在实际的应用方面，APM、Pixhawk飞控，Mission Planner、QGroundControl地面站等均使用了MAVLink协议进行通讯。

**3、OpenCV计算机视觉库**

 OpenCV于1999年由Intel建立，如今由Willow Garage提供支持。OpenCV是一个基于BSD许可（开源）发行的跨平台计算机视觉库，可以运行在Linux、Windows和Mac OS操作系统上。

**4、融合组成**

 在四轴飞行器的自动控制的软件方面，我们团队选择了基于ArduPilot团队编写的ApmCopter的开源飞控系统和基于Python版本的OpenCV的计算机视觉库。开源的ApmCopter系统对于MavLink协议的支持十分的良好。因此，使用开源的ApmCopter系统所提供的对于MavLink协议的支持，可以大幅度的简化在树莓派的程序当中，与飞控系统通信的程序部分。OpenCV计算机视觉库同样是开源的。因此，OpenCV被广泛的用于机器视觉的处理当中。而在我们的四轴飞行器的自动控制程序当中，OpenCV占据了一定的代码编写量。其中，我们团队主要将OpenCV用于将图像二值化后，将图像分区，并按引导线出现在不同分区的情况，来处理循迹部分的程序，以及任务的执行阶段部分的判断程序。

 MavLink协议和OpenCV计算机视觉库都提供了Python语言的实现，并将其封装成了相应的包。对于大幅度的降低程序的编码量，起到了重要的作用。因此，我们团队将树莓派作为硬件平台，并在树莓派当中安装了Raspbian版本的Linux系统。在系统搭建完成以后，我们对相应的Python环境和其他的一些程序所需的环境，进行了搭建。最终，通过编写了相应的程序，将软件和硬件进行了有机的融合，从而实现了四轴飞行器的自主循迹和运输物品的功能。

 另外，在调试和运行的过程中，我们使用了2.4G频段的Wi-Fi 802.11n和SSH进行控制，图像和数据的实时监控采用了2.4G频段的Wi-Fi 802.11n和VNC的组合。这样的组合，大幅度的降低了整体程序的开发难度和四轴飞行器整体的造价。

**5、主要自动控制程序的Python程序的函数定义和代码解释**

def get\_white\_precent(istart, jstart, iend, jend): #用于获取二值化以后的图像，白色像素点所占的比例。

def get\_black\_precent(istart, jstart, iend, jend): #用于获取二值化以后的图像，黑色像素点所占的比例。

def get\_block\_info(): #用于将二值化以后的图像，按照大小分块。然后计算每块的黑色像素点所占的比例，并加入到列表当中。最后返回列表给调用此函数的地方。

def draw\_block(): #用于将每个块的边界，显示在输出的图像当中。

def get\_status(num): #用于判断每个块内黑色像素点所占的比例，是否符合最小黑色像素点所占的比例的最小要求。

def next\_step(num): #用于判断飞行器下一步的移动方向。

def next\_step\_up(num): #用于判断飞行器起飞或上升时的下一步的移动方向。

def next\_step\_down(num): #用于判断飞行器降落或下降时的下一步的移动方向。

def hexShow(argv): #光流的超声波传感器的高度数据的处理。

def read\_height(): #读取光流的超声波传感器的数据。

def read\_height\_sr(): #读取HC-SR04超声波传感器的高度数据。

def get\_height(): #对读取高度的不同类型的超声波传感器进行适配。

def add\_height(step=1,skip=0): #增加四轴飞行器的高度。

def sub\_height(step=1,skip=0): #减少四轴飞行器的高度。

def north\_loiter(step=1,times=0.25): #向四轴飞行器的北方移动（推荐在Loiter模式下运行）。

def south\_loiter(step=1,times=0.25): #向四轴飞行器的南方移动（推荐在Loiter模式下运行）。

def west\_loiter(step=1,times=0.25): #向四轴飞行器的西方移动（推荐在Loiter模式下运行）。

def east\_loiter(step=1,times=0.25): #向四轴飞行器的东方移动（推荐在Loiter模式下运行）。

def height\_loiter(num,check=0): #改变四轴飞行器的高度（仅支持在Loiter模式下运行）。

四、项目特色与创新点

 本项目在技术和应用上，均具有一定的创新性。在技术上，本项目可以实现程序自主控制飞行，能够按照提前预定的程序，自动完成所部署的任务。在应用上，本项目具有一定的自主拾取物品的运输能力，能够在实际应用当中，实现物品运输的任务。

 其功能上的实现，主要通过现有的硬件和飞控系统为基础，通过将Pixhawk飞控硬件系统所运行的ArduCopter飞控程序，与树莓派所搭载的摄像头和OpenCV计算机视觉库程序相结合。其中，利用了Mavlink无人机通讯协议，用Python语言编写了相应的控制程序，将两者项结合在了一起，从而实现了无人机的自主巡线的功能。因此，本项目在能够成功完成和实现预定的目标的同时，也展示了本项目在技术和应用上，具有一定的创新性。

五、项目总结

 通过完成本次大创项目，使我们学习到了很多的知识。在完成这个大创项目的过程当中，我们主要涉及到了无人机的自动控制和视觉循迹这两个主要的组成部分。通过参与本项目，使我们在参与项目的同时，充分的提高软件的编写和设计能力。与此同时，参加并完成了本次大创项目，也为我们将来的学习和研究，打下了良好的基础。

参考文献：

[1]龚有敏.四旋翼无人机轨迹跟踪与自主着陆控制研究[D].哈尔滨工业大学,2017.

[2]陈鹏飞.基于Linux的无人机自动驾驶仪系统设计[D].南昌航空大学,2018.

[3]徐健.自主无人机飞行控制系统设计和实现[D].广西大学,2018.