**基于视觉的云台跟踪系统设计**

|  |  |
| --- | --- |
| 学 院 | 航空航天工程学部 |
| 专 业 | 探测制导与控制技术 |
| 班 级 | 1434030601 |
| 学 号 | 143403060132 |
| 姓 名 | 张子阳 |
| 指导教师 | 宋崎 |
| 负责教师 | 宫廷 |

沈阳航空航天大学

2018年6月

**摘 要**

随着人工智能的发展，图像处理技术在现代生活中有着越来越重要的影响。视觉追踪技术的发展也在一定程度上改变了人们的生活。树莓派是一种新型的控制芯片，在近几年得到了快速的发展和广泛的应用。本文利用树莓派和opencv库对摄像头采集到的视频信息进行识别和分析，得到目标的相对位置。然后根据相对位置控制云台转动，使运动目标始终保持在摄像头的画面中心，完成视觉追踪的过程。实验结果表明，该方法可以在简单背景下，有效地对运动目标进行识别和追踪，即当目标运动时，云台会跟随目标而运动。

关键词：图像处理；视觉追踪；opencv；树莓派；摄像头；云台

**目 录**

[1 绪论 3](#_Toc515269669)

[2 方案设计 4](#_Toc515269670)

[2.1 树莓派介绍 4](#_Toc515269671)

[2.2 云台与摄像头 5](#_Toc515269672)

[2.2.1 云台 5](#_Toc515269673)

[2.2.2 摄像头 6](#_Toc515269674)

[2.3 PWM波介绍 6](#_Toc515269675)

[2.3.1 背景介绍 6](#_Toc515269676)

[2.3.2 基本原理 6](#_Toc515269677)

[2.4 视觉追踪方案设计 6](#_Toc515269678)

[3 图形算法 8](#_Toc515269679)

[3.1 图像处理 8](#_Toc515269680)

[3.2 确定运动方向 8](#_Toc515269681)

[4 控制算法 9](#_Toc515269682)

[5 实验结果及分析 9](#_Toc515269683)

[参考文献 10](#_Toc515269684)

[致谢 11](#_Toc515269685)

# 绪论

随着现代科技的发展，人工智能已经成为当今社会的一个不可缺少的新技术，而计算机视觉作为人工智能的一个重要分支，已经广泛应用于信号处理、光学、应用数学等诸多领域，代替人类完成对目标的识别和分类。opencv是一个开源的、跨平台的计算机视觉库，包含了很多常用且强大的视觉处理工具，可以在商业和研究中免费使用。树莓派是一种基于ARM的微型电脑主板，以SD/MicroSD卡为内存硬盘。它在最近几年中得到了巨大的发展和广泛的应用。它的作用相当于单片机，用来处理接收到的数据，但是它的功能比普通的单片机更强大。它不仅包含IO引脚，也包含普通单片机没有的USB插口、HDMI插口和音频插口等许多通用插口，可以很方便地连接到其他外部设备上。由于树莓派采用python为主要的编程语言，所以本文的所有功能将通过python实现。本文通过树莓派和opencv来处理视频信息，并控制云台转动，实现了对运动物体的识别和追踪。

目前国内外对视觉追踪已经有了较为完善的研究成果。例如提取物体特征从而跟踪物体[1]、跟踪特定颜色的物体[2]等。总体来看这些方法都能完成对物体的识别和跟踪，但这些方法跟踪的是特定的目标，在目标的选择上有所局限。树莓派作为一种新型的控制芯片，具有操作简单、使用成本低的特点，但目前相比于其他单片机来说并未完全普及[3]。利用树莓派，可以很方便地处理图像和控制云台运动。本文利用树莓派、三轴云台和摄像头，提出了一种能够识别并跟踪所有运动物体的方法。虽然需要与目标差别较大的背景，但很好地解决了只能识别特定物体的问题。

本文在第2章介绍了视觉追踪方案所需要的硬件设备和总体的方案设计，第3章介绍了处理图像的图形算法，第4章介绍了控制云台运动的控制算法，第5章展示了最后的实验结果，并对其进行了分析。实验结果表明，通过树莓派处理图像，控制云台转动，可以做到对移动物体的视觉追踪。

# 方案设计

## 树莓派介绍

树莓派是一款只有信用卡大小的arm架构的小型计算机，可以运行windows系统或者linux系统[4]。它是由英国的树莓派基金会所开发的，在本文中运行的是树莓派官方的raspbian系统，系统的运行界面如图2.1所示。Raspbian系统是转为树莓派开发的一种linux系统，与正常的linux系统的操作方式相同。在这个系统中，预先安装了很多为树莓派提供的开发环境和编译工具，也预先安装了python3，所以本文将在python3环境下进行开发。



除了特殊的操作系统外，树莓派还有其他控制芯片没有的外观。包括40个IO引脚、4个USB接口、一个HDMI接口和网口接口等，这样的通用接口极大丰富了树莓派的功能，为开发者提供了很多方便。树莓派的外形及具体引脚的功能如图2.2所示。

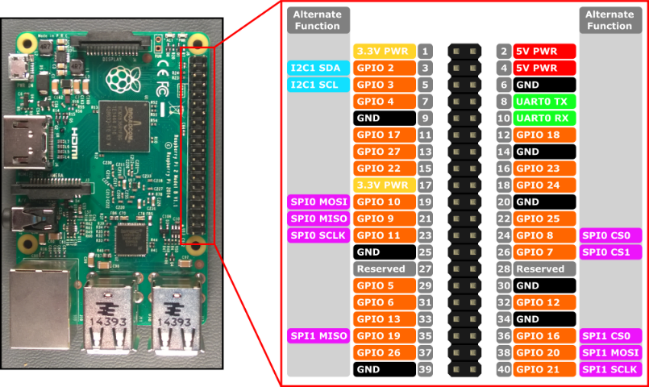


图2.2 树莓派外形及引脚功能图

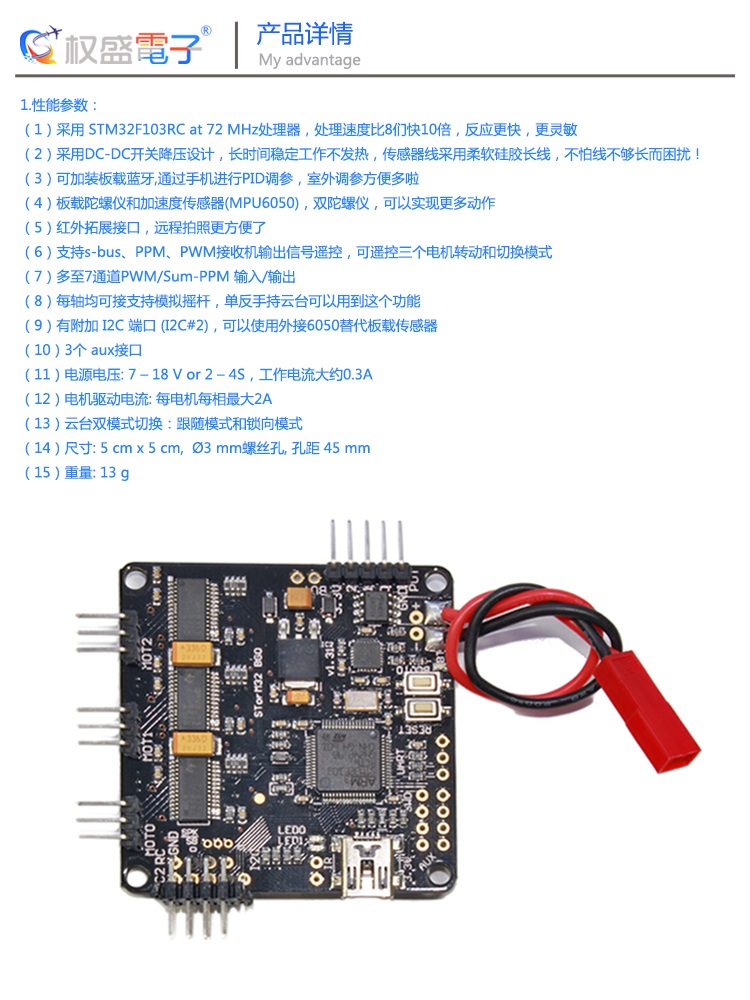
## 云台与摄像头

### 云台

云台是固定、安装摄像头的支撑设备，能够使摄像头处于人们所需要的合适位置。在云台上安装好摄像头之后，可以通过控制电机转动，改变摄像头的拍摄角度。常见的云台包括二轴云台和三轴云台。为了方便实验顺利进行，本文采用的是搭载于无人机的三轴云台，实际上只控制其中两个轴的转动。这种三轴云台体积小，控制精准方便，利用PWM波就可以对云台上的电机进行控制，通电时利用普通的航模电池通电。三轴云台的图片如图2.3所示。PWM波的使用将在本文第4章中进行介绍。



在本文中使用的云台是既能左右旋转又能上下旋转的全方位云台。水平旋转角度为0°～350°，垂直旋转角度为+90°，水平旋转速度一般在3°～10°/s，垂直速度为4°/s左右。在云台底部有一个专用来控制云台的控制芯片，云台的三个电机和陀螺仪都由这个芯片控制。芯片的实物图如图2.4所示。在芯片左侧有三个与电机相连的接口，分别标有MOT0、MOT1、MOT2，我们可以通过给下方的引脚输入PWM波控制电机转动。此芯片采用STM32F103RC at 72 MHz处理器，可以安装板载蓝牙，从而通过手机进行PID调参。电源电压在12V，可以由3S航模电池供电，工作电流在0.3A。



### 摄像头

为了实验方便，我们选用的是与云台配套的无人机专用摄像头，摄像头的实物图如图所示。摄像头自带充电电池，无需外接电源，并且开机以后自动输出AV信号，无需手动调试。摄像头的信号可以通过USB数据线与树莓派相连，传输视频信号很方便。摄像头与云台有专用的绑带，这样可以很容易地将摄像头与云台固定起来。



相机选用的型号是F68的4k相机，像素高达600万像素，有照相和摄像的功能，镜头是可调鱼眼广角镜头，可以装载微型存储卡存储图片或视频资料。另外，相机还搭载了无线WIFI模块，手机可以通过专用APP与其相连，直接控制相机。

## 视觉追踪方案设计

将相机和云台固定好，云台的控制接口与树莓派引脚相连，在本文中选用树莓派的GPIO16和GPIO17作为控制云台的引脚。相机通过数据线与树莓派相连。软件部分分为对图像的处理和对云台的控制两部分，在本文的第3章和第4章会分别对这两部分进行详细介绍。摄像头获取到图像后传到树莓派中，树莓派会根据得到的图像数据进行处理，然后控制云台转动，从而保持运动物体始终在摄像头的视野中央。各部分的连接实物图如图所示。在将目标物体移动到视野中央后，再重复以上过程，获取下一帧图像，并判断目标物体是否在视野中心。软件的流程图如图1.2所示。

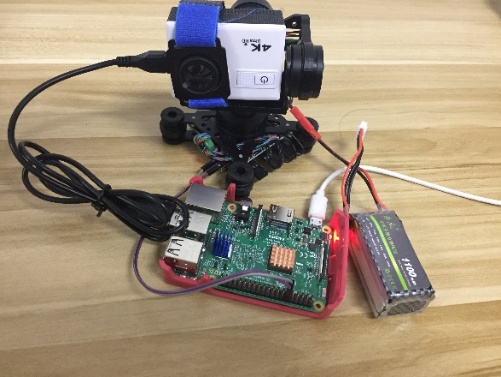


图1.1 实物连接图

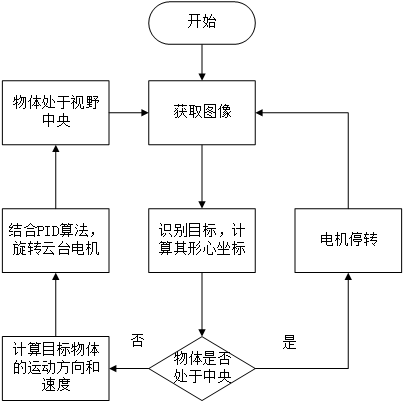


图1.2 软件流程图

# 图形算法

## opencv库介绍及其配置

opencv是由Intel公司开发的一个计算机视觉库，它由一系列的C语言函数和少量C++的类构成的，其中包括300多个C语言函数跨平台的中、高层API函数，并且不依赖于其他的外部库[5]。opencv实现了图像处理和计算机视觉的很通用的函数和算法，对非商业和商业的应用都是免费的。随着科技的发展，视觉处理在日常生活中起着越来越重要的位置，而opencv的发展使得图像处理有了一个很方便的工具。现在，opencv已经应用到了包括医疗、交通、教育等诸多方面，为我们的生活和图像处理技术提供了巨大的方便。因此，本文选用opencv库作为处理摄像头图像的工具。

在使用opencv之前，我们需要先在树莓派上配制opencv才能使用。配制时需要下载opencv库及其子库，但对于linux系统来说，默认的镜像源在国外，导致下载时间太长，所以在配置前需要更改镜像源的地址，即在命令行输入更改镜像源的命令，命令如表所示。

sudo -s

echo -e "deb http://mirrors.ustc.edu.cn/raspbian/raspbian/ stretch main contrib non-free rpi

deb-src http://mirrors.ustc.edu.cn/raspbian/raspbian/ stretch main contrib non-free rpi" > /etc/apt/sources.list

echo -e "deb http://mirrors.ustc.edu.cn/archive.raspberrypi.org/ stretch main ui" > /etc/apt/sources.list.d/raspi.list

exit

sudo apt update && sudo apt -y upgrade

在更改镜像源之后就可以对opencv进行配置了。配制opencv的方法有两种，一种是逐个下载opencv所需要的子库，然后利用CMAKE进行源码编译，这种方法过于繁琐，需要对应正确的版本逐个下载，而且源码编译时耗费时间较长，不推荐使用。另一种方法是利用树莓派中已经安装的PYTHON语言的PIP工具进行配制。PIP是专门为PYTHON开发的一个下载工具，可以很方便地下载需要的任意PYTHON的工具库。在经过数次调试之后，得知可以通过表中命令配制opencv。

pip3 install opencv-python

sudo apt-get install libatlas3-base

sudo apt-get install libjasper1

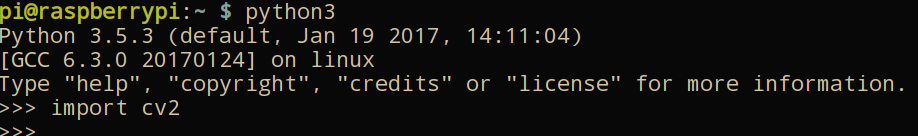
sudo apt-get install libgst7

sudo apt-get install python3-gst-1.0

sudo apt-get install libqtgui4

sudo apt-get install libqt4-test

其实使用python3后，再输入import cv2，从错误信息中就能知道缺少什么库，再使用sudo apt-cache search相关包就能查询到名称，最后使用apt安装即可。当配置成功后，在命令行中使用python3，输入import cv2进行检查，发现没有错误信息时，表示配制成功，可以使用opencv库了。配制成功时命令行的显示如图所示。



## 图像处理

将摄像头得到的信息传入树莓派中，记录当前这一帧的图像，做灰度处理和高斯模糊。如果是第一帧，就跳过后续步骤并记录下一帧的图像，不进行其他计算。在得到下一帧图像后，对这两帧图像进行差值计算，找出两帧图像的不同处，这个不同处就是移动的目标。实现找到差值的函数为absdiff(src1, src2, dst)。

为标记出运动物体的范围，需要确定差值的边界。实现找到边界的函数为findContours(image, mode, method)，而其中的输入图像image必须是一个二值化图像，所以我们需要在此之前，对差值图像进行二值化运算。为显示清晰，在原始图像上，目标的边界外围增加一个矩形边框，其实现函数为boundingRect(img)。矩形边框的中心就是目标物体的中心。这个位置与屏幕的正中心位置相比较，可以得到一个移动向量，这个向量对于之后控制云台的程序来说十分重要。

需要注意的是，没有可以直接找到窗口中间位置的函数，所以为了确定窗口的中间位置，需要在处理图像之前就先固定窗口的尺寸，相关的实现函数为resizeWindow(winname, width, height)。图形算法的伪代码如表2.1所示。

表2.1 图形算法伪代码

|  |
| --- |
| *While(1):*  *gray = cvtColor(camera video,COLOR\_BGR2GRAY)*  *blurred = GaussianBlur(gray)*  *if firstFrame is None:*  *firstFrame = blurred*  *continue*  *frameDelta = absdiff(firstFrame,blurred)*  *thresh=threshold(frameDelta,THRESH\_BINARY)*  *cnts = findContours(thresh)*  *x, y, w, h = boundingRect(cnts[0])*  *rectangle(frame, (x, y), (x + w, y + h))*  *a = int(x+w/2)*  *b = int(y+h/2)* |

为了使移动向量能够得到更清楚地展示，可以在显示目标窗口上增加一个从目标中心指向画面中心的箭头，表示下一步目标在窗口中即将到达的位置。图像处理的结果如图所示。

# 控制算法

## PWM波介绍

### 背景介绍

脉冲宽度调制（Pulse Width Modulation）简称PWM，是一种模拟控制的方式，是微处理器对模拟电路进行控制的一种非常有效的方法。目前广泛应用于测量、通信和功率控制与变换等许多领域中。

### 基本原理

## 视觉追踪的控制算法

在得到控制向量后，判断目标物体相对于界面中心的位置，就可以得到云台的转动方向。云台上的舵机可以通过PWM波控制转动。一般来说，电机有三条线，一条电源线，一条底线，一条信号线，所以将树莓派上的输出引脚与电机的信号线相连，就可以通过树莓派控制舵机。在树莓派中，输出PWM波可以利用函数GPIO.ChangeDutyCycle(duty)实现。控制算法的伪代码如表3.1所示。在第3章中，伪代码得到a和b就是目标物体的中心，假设c和d为界面的边长，e和f为控制云台的树莓派的引脚。

值得注意的是，本文所述的过程是在背景比较理想的环境下进行的，比如纯色或与目标物体颜色差异较大的环境。如果环境较为复杂，就会对所需的运动目标造成干扰，程序无法分清运动目标和背景。为解决这一问题，可以通过调试程序中的参数来解决，这些参数决定了某一步骤的具体效果。其中高斯模糊、二值化的参数尤为重要。所以，调整这些参数可以改善整个系统的效果。但这些参数的取值范围和具体数值需要考虑实际情况，并且在多次调试后对比得到，不能盲目地参考其他文献。

表3.1 控制算法伪代码

|  |
| --- |
| *def InitialGPIO(num):*  *GPIO.setmode(GPIO.BCM) GPIO.setup(a,GPIO.OUT, initial=False) a=GPIO.PWM(num,freq) a.start(0) time.sleep(1) return a*  *e=InitialGPIO(num\_1)*  *f=InitialGPIO(num\_2)*  *if a>c/2: turn right*  *else: turn left*  *if b>d/2: turn top*  *else: turn buttom* |
|  |

# 实验结果及分析

# 参考文献

1. 侯宾, 张文志, 戴源成,等. 基于OpenCV的目标物体颜色及轮廓的识别方法[J]. 现代电子技术, 2014(24):76-79.
2. 孙凯鹏, 李新军. 复杂背景下单个运动物体的实时视觉追踪算法[J]. 北京航空航天大学学报, 2008, 34(3):344-347.
3. 朱轶, 曹清华, 单田华,等. 基于Android、树莓派、Arduino、机器人的创客技能教育探索与实践[J]. 实验技术与管理, 2016, 33(6):172-176.
4. Richard Blum, Christine Bresnahan. 树莓派Python编程入门与实战[M]. 人民邮电出版社, 2015.
5. 乔·米尼奇诺, 刘波, 苗贝贝,等. OpenCV 3计算机视觉:Python语言实现[M]. 机械工业出版社, 2016.

# 致谢