

课程论文

课程名称：交互式媒体原理

开课学期：2023-2024学年第2学期

专 业：软件工程

年级班级：2022级01/02/03/04班

学生姓名：朱昊

学 号：222022321062008

指导教师：韩先锋

计算机与信息科学学院 软件学院

# 作品概述

## 创意来源及产生背景

源于对现实世界挑战的深刻洞察与对科幻未来无限憧憬的双重驱动。团队成员都亲眼目睹发生在学校旁的缙云山火，这场灾难激发了我们利用科技创新为社会带来积极变化的决心。同时，作为科幻电影的热爱者，许多电影中展示的先进人机交互技术令我等久久难忘。在这样的背景下，我们提出了“影随飞”项目——一款一体化的智能无人机控制系统。它在保留无人机自主性和灵活性的同时，更基于先进的姿态估计算法，赋予了无人机精确的姿态控制和目标追随能力。

## 主要功能与特色

1. 自动目标追随：无人机自动识别并跟踪人类目标，还可开启追随模式。

2. 姿态估计控制：通过人体的姿态来控制无人机飞行，将在第四章解释。

3. 多种操控模式：支持姿态控制/虚拟按钮两种操控模式

4. 高清视频传输：支持高清视频实时传输，能够清晰地获取目标动态。

5. 易于操作：用户界面友好，简化操作流程，初学者也能快速上手。

6. 遵循开闭原则设计：系统耦合度低，可拓展性强。

7. 环境适应性强：在大部分强度的光照下稳定工作，有较强的环境适应性。

8. 快速部署：无人机和控制系统的部署简单快捷，便于迅速进入工作状态。

## 用户群体、应用价值与推广前景

“影随飞”系统由于其人体目标跟随和姿态控制的特性，能够为广泛的群体提供显著应用价值，并拥有广阔的推广前景。随着技术的不断发展和市场需求的增长，系统有望在多个领域得到广泛应用。这其中包括但不限于以下几类：

1，航拍摄像爱好者：无人机能自动跟随拍摄对象，捕捉爱好者想要的画面

2，户外运动爱好者：进行跑步、骑行等活动时，无人机能够自动跟随记录。

3，安全监控团队：在需要长时间跟踪特定目标的场合使用无人机进行监控。

4，教育和培训：作为教育工具，激发学生对无人机技术的兴趣，促进STEM教育，教授学生无人机操作和编程。

随着无人机技术的普及和价格的降低，更多的爱好者和专业人士可以接触并使用该系统，再者随着视频内容创作的兴起，对于高质量自动跟随拍摄的需求不断增长。相信随着社会的不断进步，该系统将会有更加广阔的前景等待发掘。

# 问题分析

## 问题来源

中国已经将无人机产业的战略地位提升至国家层面，无人机与人工智能（AI）技术的结合正成为推动经济增长和产业升级的关键动力。《关于加快场景创新以人工智能高水平应用促进经济高质量发展的指导意见》等文件均明确指出了AI在无人机领域的战略应用，鼓励通过场景创新推动人工智能技术升级和产业增长，鼓励科研机构和高校在无人机与AI技术结合领域进行科研创新，推动科技成果的转化应用。

随着经济社会的发展，对于无人机在农业植保、林业监控、物流配送、公共安全等领域的应用需求不断增长。这些需求更凸显了无人机与人类高效协同工作的重要性。而近年来，计算机视觉和机器学习领域取得了显著进步，尤其是深度学习在图像识别和姿态估计技术上的重大突破。这些技术进步为无人机的智能化提供了强有力的算法支持，使得无人机也能够精准地识别和响应复杂的环境和人体姿态，从而实现更加智能化和自动化的作业流程。

## 现有解决方案

天津大学的李宏跃老师曾执导过一个创新训练项目[[1]](#footnote-0)，题为“基于嵌入式计算机的无人机体感交互控制器”。其项目通过人体姿态识别技术，为无人机控制提供了一种新颖的交互方式，并通过实际测试验证了其有效性和实用性。我们项目的部分想法与其如出一辙却又标新立异，下面的表格很好的说服了这一点。

表2-1 解决方案同异分析

|  |  |  |
| --- | --- | --- |
|  | 天津大学项目 | 本项目 |
| 姿态估计算法 | 使用OpenPose | 使用MediaPipe |
| 姿态控制功能 | 只能识别简单的姿态来对无人机进行简单的控制 | 能够识别较复杂的姿态，还可以对无人机进行抛飞，手掌降落等复杂控制 |
| 目标追随功能 | 未实现该功能 | 成功实现 |
| 识别准确度 | 91.33% | 95.375% |
| 帧率 | 低于10fps | 稳定在30fps左右 |
| 可扩展性 | 未设计 | 遵循开闭原则，耦合度低，拓展性强 |
| GUI框架 | 未明确指出 | 使用PySide2 |
| 使用设备 | 外接摄像头+无人机+Jetson Nano嵌入式平台 | Tello无人机 |
| 系统部署 | 部署在NVIDIA Jetson Nano嵌入式平台上 | 部署在Windows桌面平台 |
|  |  |  |

综上比较，本系统毋庸置疑不仅选择了更加轻量化和先进的姿态估计算法，而且对此算法的应用进行了更加深入的研究，最终展示出来的就是一个集丰富功能、易于操作且稳定美观于一体的无人机控制系统

## 本作品要解决的痛点问题

1.无人机的基础控制问题：

如何利用现有API向上层提供对无人机起飞、悬停、降落等基本飞行 动作的控制接口。

1. 优化视觉处理方案，实现预期功能问题：

确定何种视频传输方式

选择何种姿态估计算法以兼顾视频帧率，视频实时性等问题

如何决策以实现姿态的识别和对无人机进行控制

如何决策以实现无人机目标追随功能

3.用户界面（UI）开发问题：

如何设计欢迎界面，使用户一键连接无人机

如何设计控制界面以整合虚拟按钮，视频参数和无人机参数等于页面中

如何加入音效元素，以提升用户反馈

4.系统的可扩展性问题:

如何开发无人机抽象类

如何优化代码结构，以实现模块间的解耦

## 解决问题的思路

为了解决本项目中提出的痛点问题，我们采取了以下思路：

1. 视觉处理与姿态估计：选择适合的算法和工具来实现高准确度的人体姿

态估计。我们考虑使用 MediaPipe 作为姿态估计算法，因为它在准确性和实时性之间提供了良好的平衡。

2. 目标追随与姿态控制的决策逻辑：基于姿态估计数据，利用不同关键点 的距离和角度来设计算法以识别人体和控制无人机的飞行行为。这将涉及到对关键点的识别和跟踪，以及要结合这些数据和 PID 控制器来对无人机发送飞行指令。

3. 系统的可扩展性：确保系统设计允许未来集成新的无人机模型和功能。

我们将采用遵循开闭原则的模块化设计，并通过顶层抽象类 AbsDrone 来实现这一点。

4. 测试与验证：在开发过程中，我们将进行单识别姿态准确度测试、无人

机控制功能测试和软件系统配置要求测试，以确保所有功能按预期工作，并在第

5 章中报告测试结果。

通过上述思路，我们期望能够实现一个功能全面、操作简便、可扩展性强的

影随飞--智能无人机控制系统，以满足不同用户群体的需求，并在实际应用中展 现出其价值。

# 技术方案

## 技术路线

想实现预期的功能实属不易，在经过团队成员对数篇文献、网上资料和灵感想法的抽丝剥茧整理汇总后，最终敲定按下图所示的技术路线来实现我们整个无人机飞控系统。

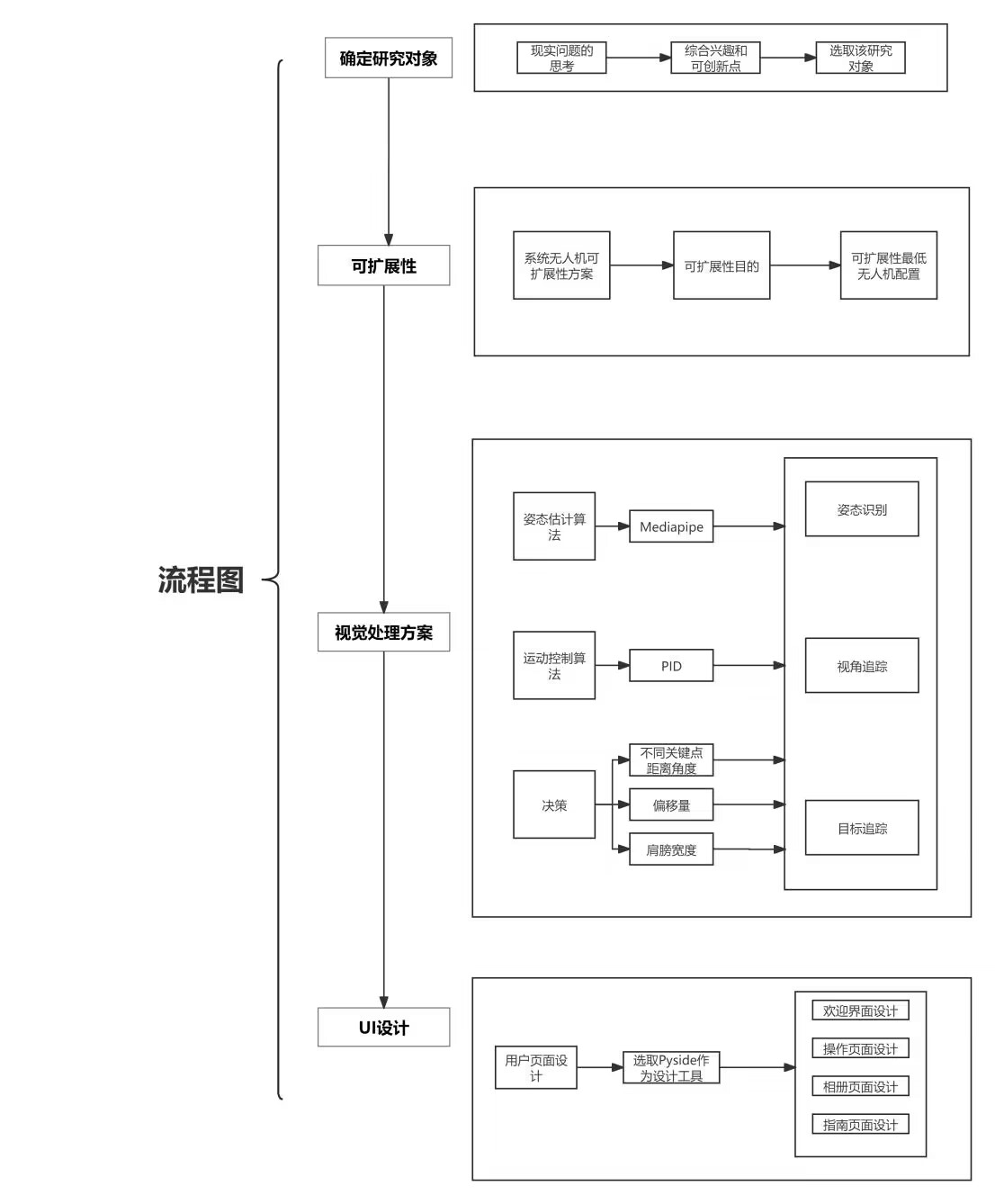


图 3-1 技术路线图

在深入讨论并明确要实现的功能后，我们首要的任务是对代码的结构进行预估分析以满足对可扩展性的要求。随后从多重因素考虑制定无人机的选取原则以及使用本系统的最低配置。基于系统稳定性和实时性的考虑，最终选择MediaPipe的Pose Landmark detection方案来作为我们的姿态估计算法[1]。通过检测视频中心点和人体追踪点的偏移量来反应目标偏移状况并提供依据。计算算法识别到的不同关键点的坐标之间的距离和角度作为依据来识别姿态。目标追随则通过计算人体两个左右两个肩膀关键点的距离来作为决策依据。以上三种依据结合PID控制器以及底层封装好的飞行控制接口就可实现对应的功能。最后利用PySide2来实现用户界面的开发，再结合音效来增强用户反馈。

## 可扩展性和无人机飞行控制方案

### 系统无人机可扩展性介绍

为了增强系统在不同用户环境下的适应性，我们团队希望使系统能够兼容多种类型的无人机，即实现系统无人机的可扩展性。这种可扩展性对于用户来说非常有价值，使其能够在多种场景和条件下部署无人机。同时一个支持多种无人机的系统可以吸引更广泛的用户群体，用户可以使用现有的无人机设备，而不需要专门为系统购买特定型号的无人机，这有助于增加潜在客户的数量，提高产品的市场竞争力。

### 本系统的无人机的最低配置要求介绍

### 基于无人机可扩展性的理念，为了确保系统能够适配并有效地控制各种无人机，我们为用户提供了一套清晰的最低配置要求。这些要求将帮助确保所有适配的无人机都能够支持基本的功能和性能标准，从而保证系统的正常运作和安全。以下是建议的无人机最低配置要求：

1.控制接口兼容性：

无人机应具备开放式或已知的编程接口（API），允许外部控制其基本飞行操作，如起飞、悬停、降落等。

2.通信能力：

无人机需要具备稳定的无线通信能力，至少支持Wi-Fi连接，以便实时传输控制指令和接收数据。

3.摄像头和视频传输功能：

至少配备一个前置摄像头，能够进行视频捕捉，并支持实时视频流传输，用于系统的视觉处理和飞行监控。

4.传感器配备：

必须装备必要的飞行控制传感器，如加速度计、陀螺仪、GPS（用于位置定位和稳定飞行），这些是执行基本导航和稳定控制的关键组件。

5.电池和续航能力：

需要具备足够的电池容量，至少能够支持15分钟以上的连续飞行时间，确保可以完成一定时长的任务操作。

## 视觉处理方案

### 姿态估计算法的选择

人体姿态估计是计算机视觉领域中的一个重要研究方向，其主要目标是从图像或视频中识别出人体的姿态信息[5]。如今许多人体姿态估计模型都已经开源，如在第二章的现有解决方案提到的天津大学李宏跃老师指导的项目中利用的就是美国卡内基梅隆大学Zhe Cao等人在2018年提出的OpenPose模型[9]。对比近几年流行和广泛使用的人体姿态估计模型，我们可以得到表3-1。

表 3-1 不同姿态估计算法的对比

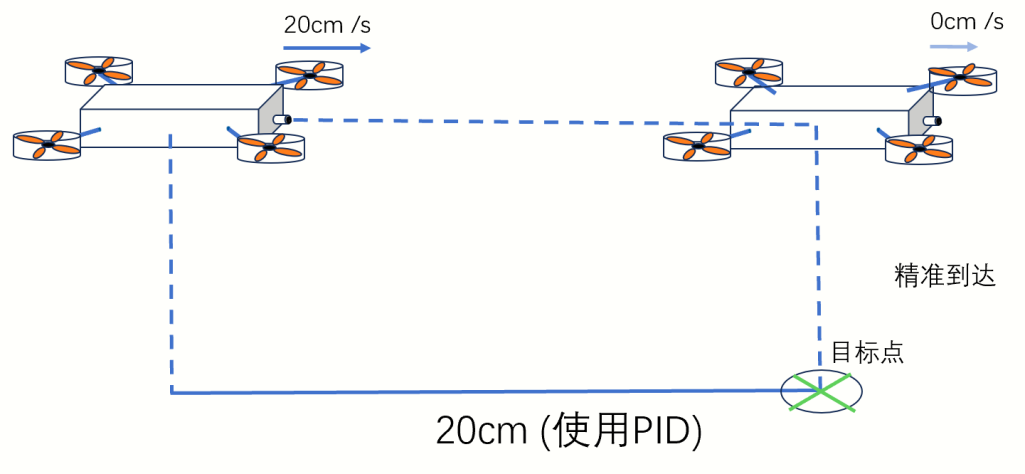
|  |  |  |  |
| --- | --- | --- | --- |
|  | OpenPose[9] | AlphaPose[6] | MediaPipe[7][3] |
| 开发者 | 卡内基梅隆大学 | 上海交通大学 | Google |
| 支持的骨架关键点数量 | 18 | 18 | 33 |
| 易用性 | 依赖特定的库或环境，配置极其复杂 | 依赖特定的库或环境，配置较复杂 | 不需要配置，只需要导入包即可，API简单易用 |
| 准确性 | 高 | 略好于OpenPose | 略低于OpenPose |
| 实时性 | 计算量极大，帧率较低（10fps左右） | 计算量较大，帧率中等（20fps左右） | 轻量化模型，计算量降低，帧率较高（30fps左右） |
| 更新及维护频率 | 低 | 不规律 | 高 |
| 硬件要求 | GPU要求极高 | GPU要求极高 | CPU就可以轻松达到预期 |
|  |  |  |  |

本系统作为一个要与用户进行实时交互的飞行控制系统，帧率低导致的控制时延是无法忍受的。而Google公司提供的MediaPipe pose解决方案只在牺牲较小精度的情况下就可以给帧率和延迟带来质的提升[1]。不仅如此，其配置简单、易于使用、跨平台性能好以及易集成的特性也是吸引我们团队使用的重要因素。

### PID控制算法的选取(本人负责)

在无人机的姿态控制和视角跟踪功能中，我们采用了PID控制算法来精确调节无人机的飞行行为[8]。常规的恒速控制指令难以实现精确的定位和悬停。例如，当无人机以5cm/s的速度飞行至20cm目标距离时，由于惯性，它往往会超出预定位置。通过引入PID控制算法，无人机能够逐渐减速，直至完全停止，确保精确到达并悬停在指定位置。

为了实现这些控制功能，我们选用了开源的simple-pid算法，它不仅简单而且效率高，非常适合在无人机控制系统中应用。这种算法的优势在于它能够快速调整无人机的飞行状态，响应各种飞行指令，从而实现高精度的动态控制。



### 视角跟踪功能的实现原理

视角跟踪功能就是使无人机的飞行转向（正前方）始终朝着目标的方位，飞行高度始终与目标高度齐平，保证目标始终处于无人机摄像头拍摄画面的中央位置。这是姿态控制功能和目标追随功能的基础，因为只有保证人物始终出现在画面中央，系统才能对人物进行姿态估计并对无人机进行指令控制。具体实现原理如下：

Step1.获取画幅宽度和高度：获取无人机摄像头拍摄画面的宽度和高度分别记为w和y。

Step2.目标跟踪点的标定：使用MediaPipe姿态估计算法标定人物关键节点[4]。按照优先级，“鼻子”、“脖子”和“臀部中心”依次被选为目标跟踪点。如果“鼻子”节点可检测，则作为跟踪点；若不可检测，选择“脖子”，如果“脖子”也未检测到，最后选用“臀部中心”，记为T。

公式1

Step3. 无人机拍摄画面中心点的标定：标定距离画幅原点0.5w和0.35y的位置作为无人机拍摄中心点，记为C。

公式2

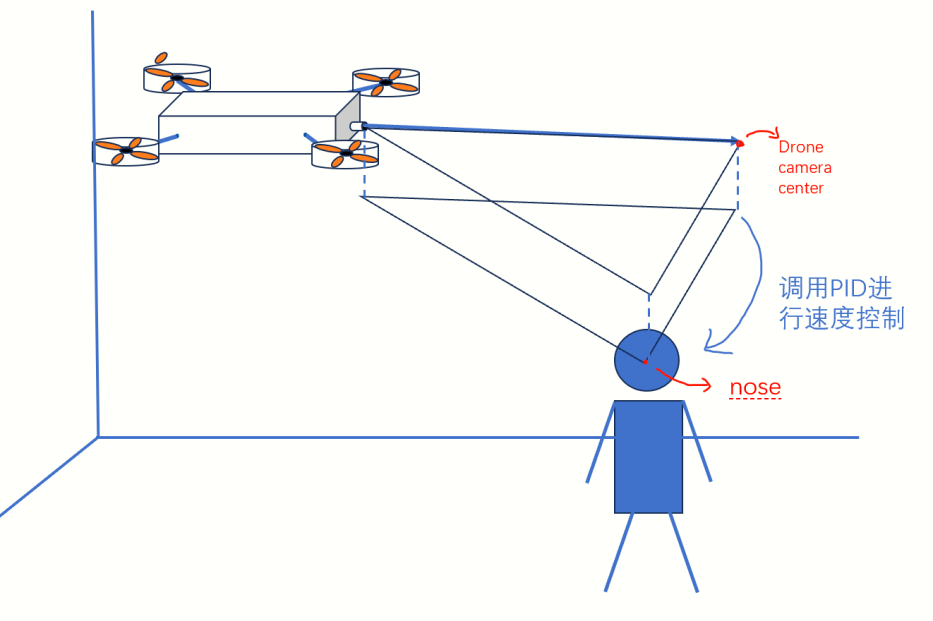
Step4. 调用PID实现无人机视角跟踪：在目标跟踪点和无人机中心点标定后，计算出水平偏差和垂直偏差。

公式3

调整PID参数，得到（x）和（x）函数，调用PID函数分别对每一帧画面获取得到的和进行计算，即可得到该时刻对应的无人机旋转速度 和垂直飞行速度。

公式4

这样，系统会将每一帧图片对应的和实时发送给无人机以实现无人机视角跟踪的效果。



### 姿态识别与控制实现原理（本人负责）

#### 姿态关节点的获取与计算

MediaPipe pose对人体各个关节位置进行估计，将其作为整体图像，并利用神经网络进行特征提取[3]。MediaPipe会对每个关键点的位置进行计算，再将其转化为的二维的归一化坐标（以图片的左上角为原点）。识别到的人体关键点图如图3-2所示。

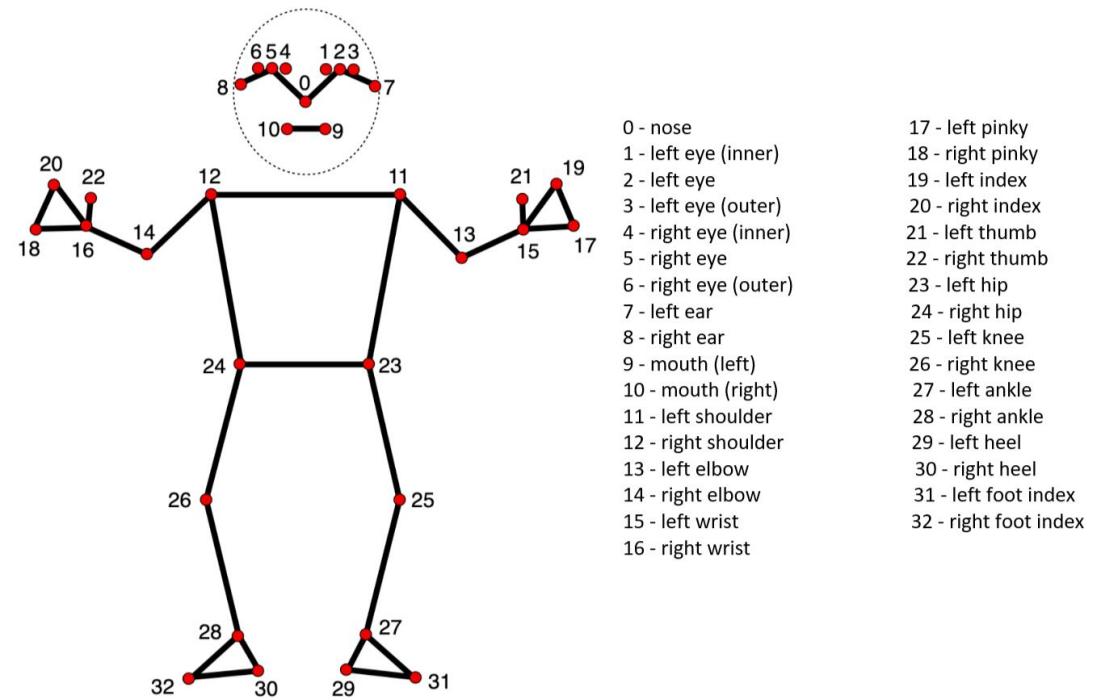


图3-2 关键点说明

在MediaPipe的python实现中，33个关键点的坐标会存储在检测器的一个二维列表中[1]。将该列表乘以视频的分辨率读取到我们的系统中，就可以利用这些数据来进行计算和决策以识别相应的动作。由于本项目设计的姿态控制姿势都是上半身的姿态动作，所以着重计算的是上半身关键点的相对关系。为了实现项目的需求，还有两个关键点的坐标是必不可少的。

一个是neck，其为left\_shoulder 和 right\_shoulder 的中心点。

另一个是hip，其为left\_hip和right hip的中心点。

计算公式如图所示（注意：第三章全部以关键点名\_x和关键点名\_y的变量均为对应关键点的x轴坐标数值和y轴坐标数值）

关键点之间的距离和角度是决策关键点之间的相对位置是何种姿态的重要依据。函数计算的是两个关键点的欧式距离。

*=*

函数计算的是向量沿顺时针方向旋转度到向量，*a*记为为函数的返回值。函数的定义如下。

基于函数，我们给出函数的定义。

#### 每个姿态对应的位置关系

计算出正确的数据后会让系统决策姿态更加如鱼得水。设想一下在现实生活中我们是如何判断“右手摸左耳”这个动作的，通常是像图3-4所示一样，通过层层的判断来确定实现了该动作。

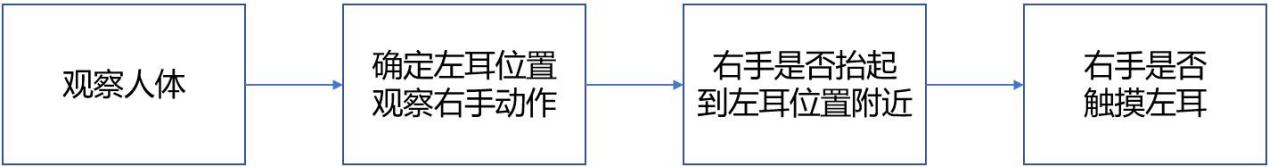


图3-4 现实中判断“右手摸左耳”流程图

我们现在的系统有了人体姿态关键点的坐标，以及判断不同关键点距离和角度的“手段”，那么我们就可以在计算机上模拟现实中的判断过程。如图3-5一样。

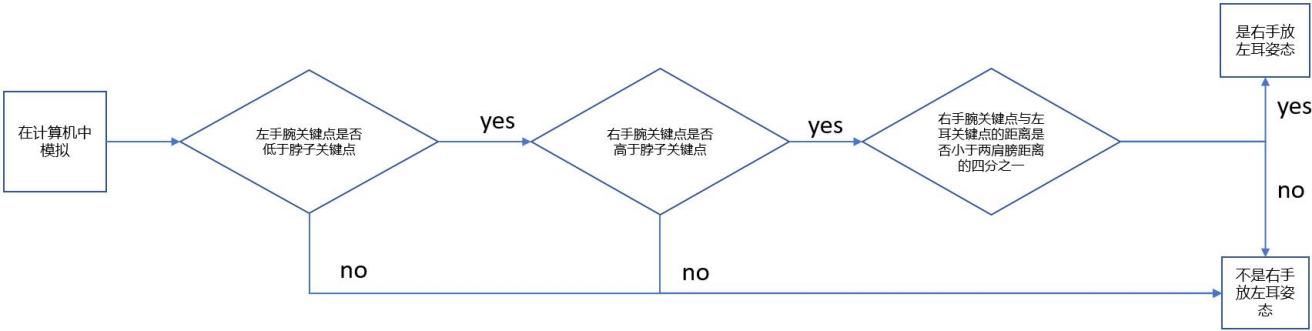


图3-5 在计算机中模拟判断“右手摸左耳”流程图

其他姿态的判断思路与这个例子如出一辙。本系统对姿态关键点做出决策的判断依据表如表3-6所示。表中利用不同关键点的坐标数值以及上述提到的两个计算函数来表示各个姿态的判断依据。需要注意的是，由于二维坐标轴的原点是视频的左上角，所以部分判断关系与直觉是相反的。为了清晰呈现表3-6，这里提前声明一个变量来表示肩膀距离

表 3-6 各个姿态动作的满足条件

|  |  |  |
| --- | --- | --- |
|  | 判断条件 | 对应姿态 |
| 1 |  | 双手交叉过头顶 |
| 2 |  | 右臂张开 |
| 3 | (4) | 右臂内收 |
| 4 | (4) | 左臂张开 |
| 5 | (4) 50 | 左臂内收 |
| 6 |  | 左手摸右耳 |
| 7 |  | 右手摸左耳 |
| 8 | 4 | 双手交叉于脖子前 |

#### 每个姿态对应的无人机动作

系统会循环读取无人机的飞行控制状态，然后传送给PID控制器中来生成下次的飞行参数并再传回给无人机[8]。正是因为有这样的循环反复向底层的飞行控制接口传递参数，才确保了无人机动作实施时的稳定和精确。

本系统中，每个人体姿态都对应一个无人机动作，如下表3-7所示。

表 3-7你把 姿态动作与无人机动作对应表

|  |  |  |
| --- | --- | --- |
|  | 人体姿态动作 | 无人机动作 |
| 1 |  | 双手交叉过头顶 |
| 2 |  | 右臂张开 |
| 3 |  | 右臂内收 |
| 4 |  | 左臂张开 |
| 5 |  | 左臂内收 |
| 6 |  | 左手摸右耳 |
| 7 |  | 右手摸左耳 |
| 8 |  | 双手交叉  于脖子前 |

### 目标追随功能实现原理

### 目标追随功能的实现基于视角跟踪功能的实现，在保证无人机视角始终跟踪目标人物后，我们只需保证无人机与人物之间的距离在一个恒定值即可实现追随效果，我们称之为距离锁定模式，即用户可以在视角追踪模式下通过一个xx姿态来开启或关闭无人机的距离锁定模式来使无人机实现距离跟踪效果。我们用肩膀距离S来映射当前无人机距离目标人物的距离，我们用映射关系来表示这个对应关系，具体数值对应关系，我们不好通过直接计算，但是我们可以肯定，与呈负相关，即肩膀距离越大，当前距离越小。

打开距离锁定模式：

1.获取锁定肩膀宽度和锁定距离：在姿态估计模式下获取目标人物关节点后，我们通过距离计算公式

*=*

传入左肩坐标和右肩坐标，获得锁定肩膀宽度,和锁定距离：

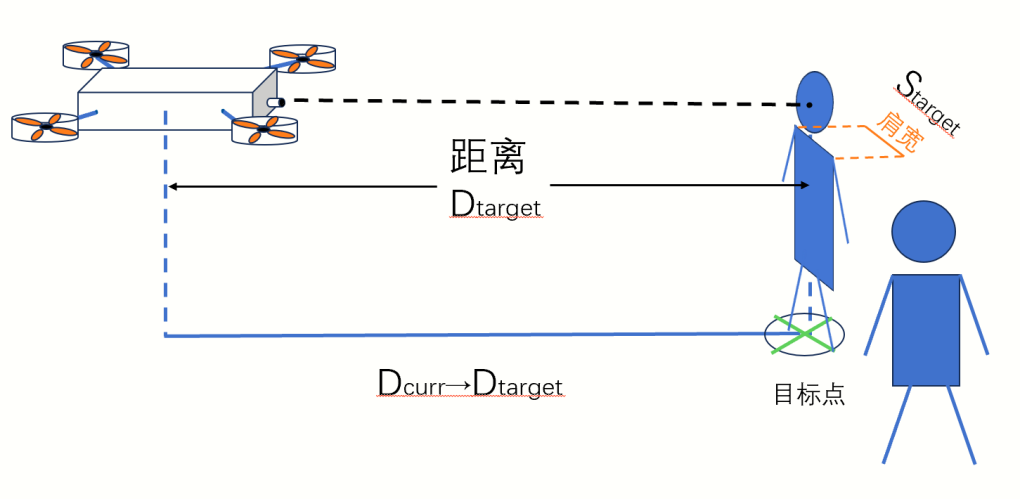
2.获取当前肩膀宽度和目标距离：在距离锁定模式开启后，同1的原理，我们计算出当前肩膀宽度和，

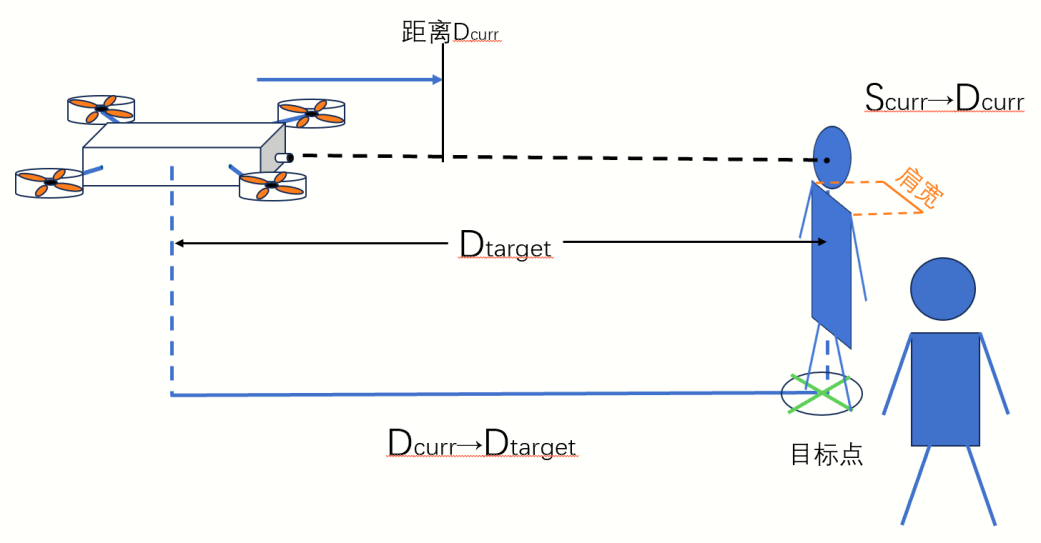
1. 调用PID实现距离锁定效果：计算当前肩膀宽度与锁定肩膀宽度的距离差，

=

我们调整参数设计控制前后距离的PID算法,记为,将传入函数即可得到当前该时刻对应的无人机前后速度。

这样，系统会将每一帧图片对应的实时发送给无人机以实现无人机目标追随的效果。





## 用户界面UI开发方案

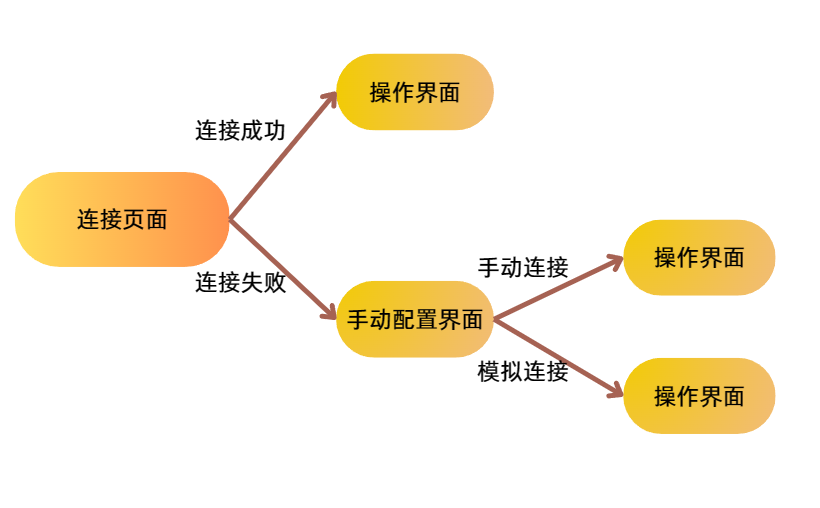
### 设计工具的选取（Qt）

选择PySide2作为系统UI界面的设计工具。原因是其具备优良的跨平台兼容性，支持Windows、MacOS、Linux等主流操作系统。同时其良好的文档和社区支持，有助于解决开发中遇到的各种问题。

由于本无人机控制系统基于python开发，调用了大量图像处理和机器学习python开源库，PySide2与python的兼容性使我们能够在同一环境中用python进行复杂的后端逻辑处理和前端界面开发，从而简化了开发流程。

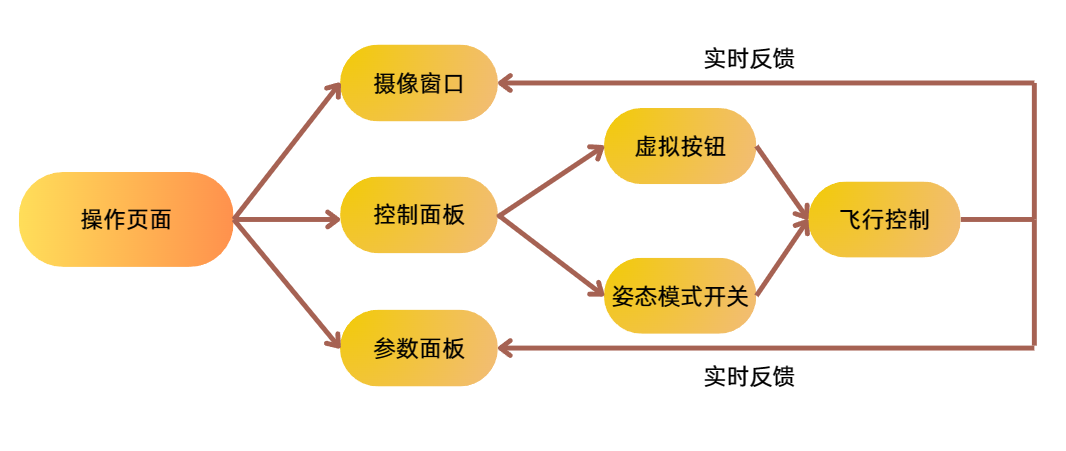
### 欢迎界面的开发逻辑

欢迎界面的核心开发逻辑是使系统与无人机完成自动连接以简化用户的前期配置操作，若自动连接失败则进入手动配置界面，在此界面用户可以进行手动配置无人机连接，若用户不具备无人机的硬件条件那么也可以选择使用计算机的摄像头进行模拟连接。开发逻辑流程图如下：



流程图2 欢迎界面

### 操作界面的开发逻辑

用户完成连接后，即可进入操作界面，操作界面的核心开发逻辑即实现一体化的无人机控制：通过虚拟按钮或者键盘完成对无人机的飞行控制、提供姿态控制开关使用户通过姿态对无人机完成飞行控制、提供无人机摄像窗口实时反馈无人机拍摄画面、提供无人机参数面板实时反馈无人机动态参数。开发流程图如下。

流程图3 操作界面

# 系统实现

## 可扩展性和无人机飞行控制实现

### 无人机实验对象的选取

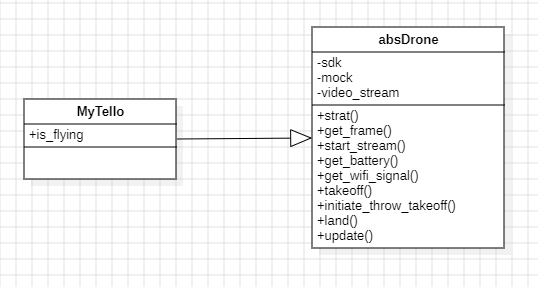
在测试中，我们需要一台特定的无人机设备作为实验对象进行开发和测试，我们选用大疆Tello轻量级教育无人机作为实验对象。大疆提供了Tello无人机的基于Python的官方SDK，开发人员可以通过SDK直接调取Tello无人机的控制函数，实现对无人机的飞行控制和视频流的实时传输，极大程度地便利了我们在实验时对无人机的基础控制，提升了开发效率。



### 无人机顶层抽象类的开发以实现可扩展性

为实现系统无人机的可扩展性，我们运用面向对象的程序设计思想，设计了顶层抽象类AbsDrone，用于封装与系统兼容的无人机控制命令。通过这种方法，开发者可以通过继承AbsDrone并实现其抽象方法来支持新型无人机，无需改动现有架构或接口。同时，我们采用模块化设计，将系统功能分解为独立协作的模块，每个模块承担特定功能。这种设计提高了代码的重用性和维护性，降低了模块间的耦合，简化了模块交互，便于模块的替换或升级。这些设计原则增强了系统的灵活性和可维护性，为未来发展打下了坚实的基础。

UML类图如下所示：

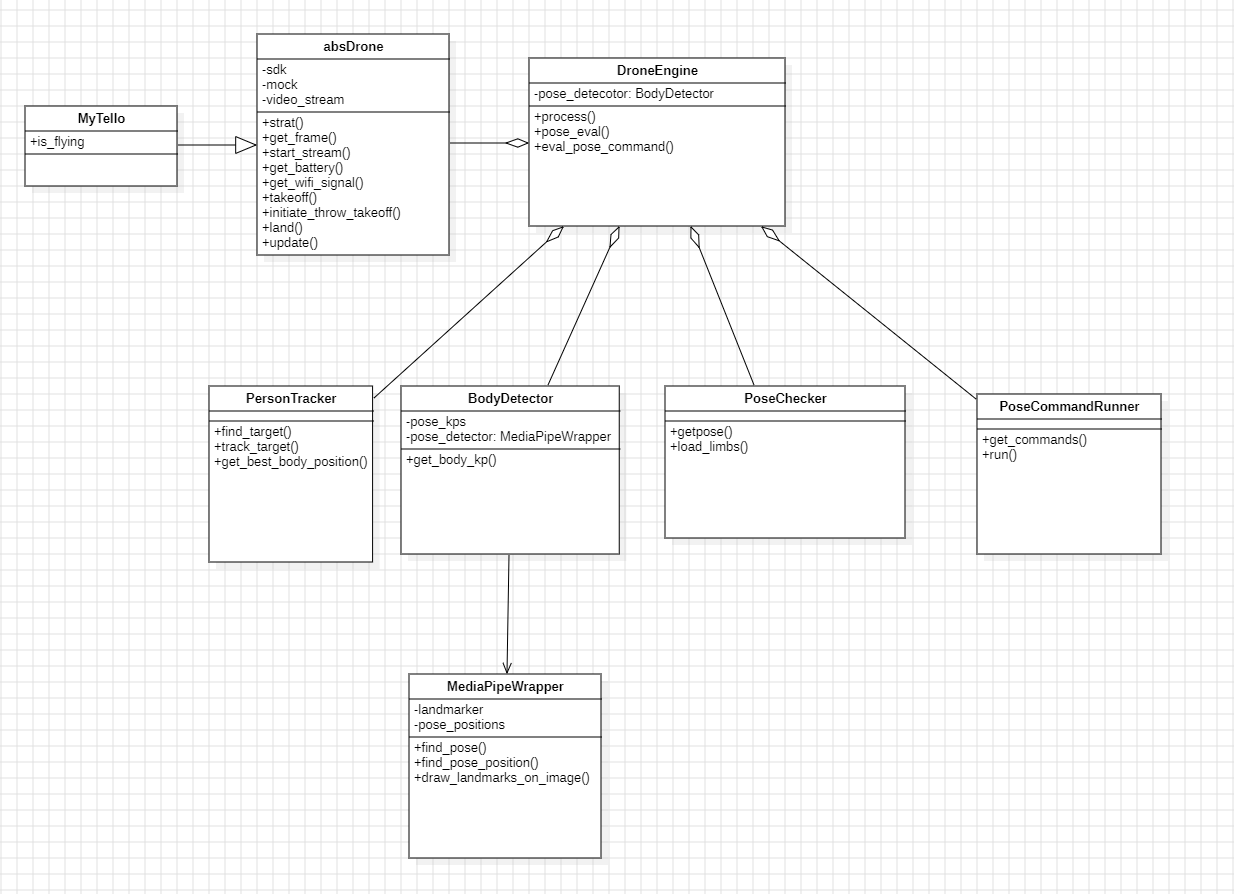


无人机的飞行控制主要通过update()函数来实现，传入四个角度的速度分别代表无人机的roll(即左右)，pitch（前后），throttle（上下），yaw（自旋）速度，这样就可以通过控制无人机四个角度的速度的方式来控制无人机的飞行。

## 视觉处理方案实现

### 视觉处理方案整体实现架构

整个系统在python平台上开发，利用python语言的高度灵活性，我们采用面向对象的程序设计方案来架构整个系统，其中视觉处理方案的UML核心类图如下所示：



我们设计了视角追踪功能，通过整合MediaPipe姿态估计和simple-pid的PID算法于PersonTracker类中。DroneEngine分析无人机传输的图像中的人体关键节点，并通过PersonTracker调用PID算法[8]，计算目标跟踪点与画面中心点的偏差，从而得出无人机的Vyaw和Vthrottle调整值，并实时更新无人机的四个方向的速度。启用锁距模式后，系统还可以通过测量和传入肩膀距离至PID算法，计算Vpitch值，实现目标追随功能。

整个项目高度面向对象模块化，每个模块负责特定功能。这种模块化不仅促进了代码的重用和维护，还保持了低耦合性，使得单个模块的替换或升级变得更加容易。

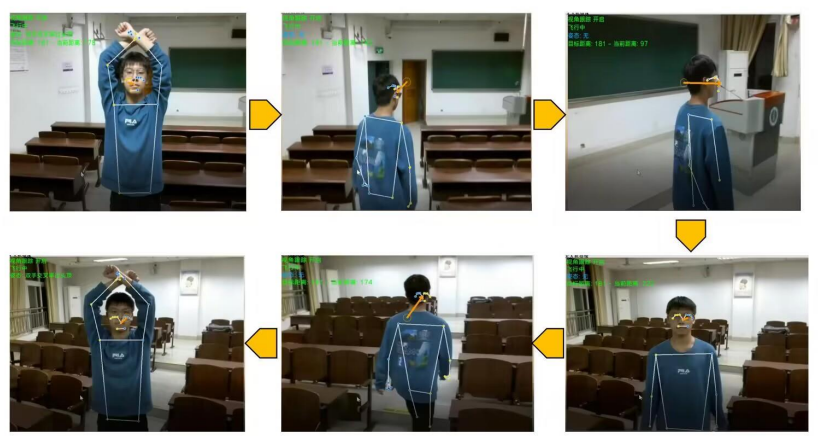
### 无人机视角跟踪和目标追随功能展示（本人负责）

通过开发MediaPipeWrapper类来封装了MediaPipe姿态估计算法的工作细节，开发BodyDetector类专门用于获取目标人物关节点，并为每个关节点赋予名称。

利用MediaPipe姿态估计和simple-pid的PID算法设计了视角追踪功能，封装在PersonTracker类中。DroneEngine通过实时分析无人机传输的图像中的人体关键节点，调用PersonTracker对象的方法，使用PID算法计算目标跟踪点与画面中心点的偏差，从而计算出无人机应调整的Vyaw和Vthrottle值，并即使发送这些控制命令给无人机，完成视角追踪的功能实现。打开锁距离模式后还可以通过标定肩膀距离，和不断传入当前肩膀距离至PID中来获得无人机应调整的Vpitch值，并及时发送这些指令给无人机，完成目标追随的功能实现。

视角跟踪实现效果如下：

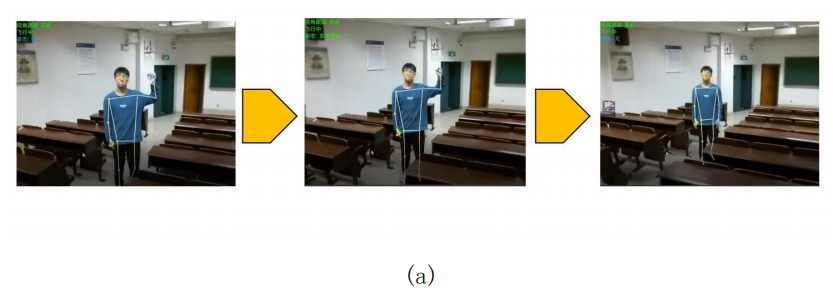
目标追随功能实现效果如下：

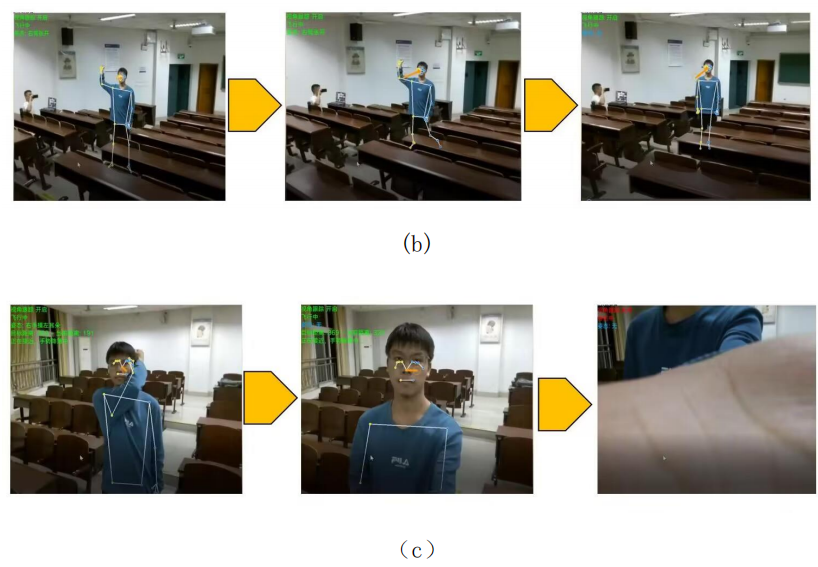


### 姿态识别与无人机姿态控制实现

依据第三章讨论的姿态识别原理，我们开发了几个关键类来实现系统功能。MediaPipeWrapper类封装了MediaPipe的姿态估计算法；BodyDetector类用于获取图像中关键节点的相对坐标；PoseChecker类负责识别姿态；PoseCommand类根据识别的姿态发送飞行指令。在DroneEngine中，这些类协同工作，处理每一帧图片以实现姿态识别和无人机的姿态控制。

实现效果如下：





## 用户界面UI实现

### 欢迎界面的实现

基于欢迎界面的开发逻辑，我们利用UI设计工具PySide2开发了欢迎界面，利用Pywifi库实现了对无人机的连接。

打开程序后首先会自动检测当前计算机是否已经连接无人机，若连接则会自动进入操作界面。



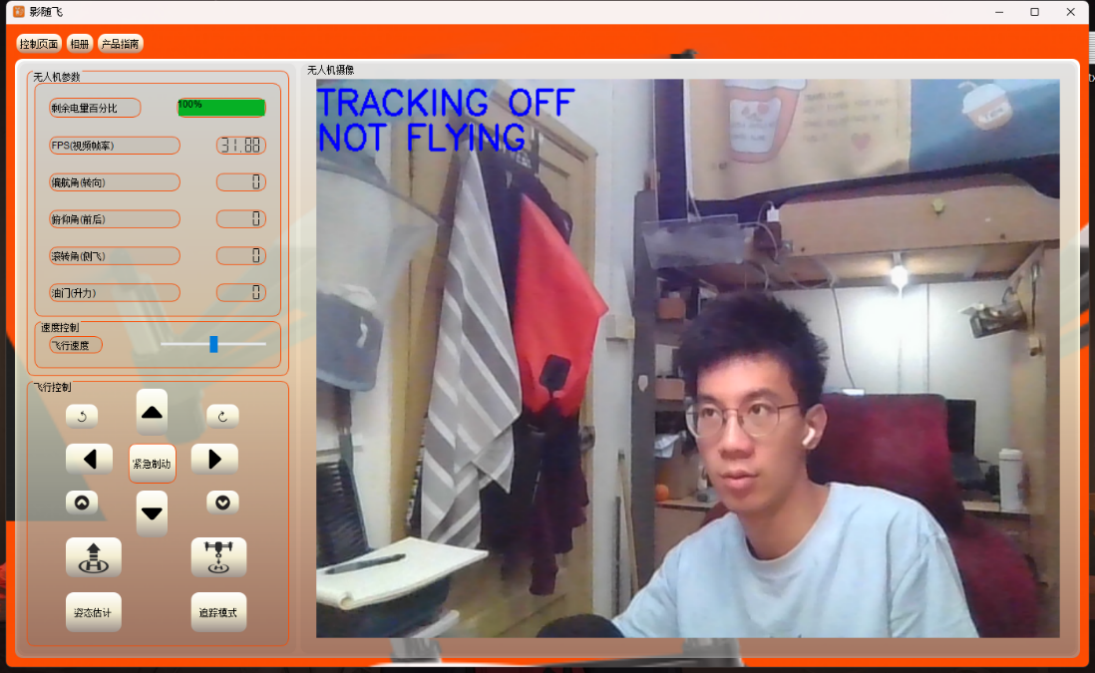
若检测到用户没有连接无人机则会提示用户进行手动配置，手动配置界面如下，点击“检测”按钮即可重写检测附近是否存在无人机wifi信号，选好无人机后点击“连接”按钮即可连接无人机。



如果用户没有无人机，那么也可以点击“模拟飞行”按钮使用计算机摄像头进行模拟飞行。

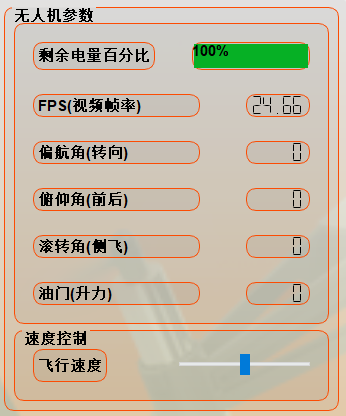
### 控制界面的实现

通过“模拟连接”进入操作界面后，进入如下界面：



左上角为菜单栏的选取，系统左侧为无人机参数和操作面板：



用户可以选择常规的鼠标键盘无人机飞行控制方式，也可以点击“姿态估计”按钮来启动姿态估计模式，使用姿态来控制无人机。

### 相册界面的开发

由于系统集成了拍摄功能，所以对于拍摄照片的存储显得尤为重要，拍摄照片默认存储于项目文件夹的assets/pictures文件夹下，同时在前端提供了相册页面供用户浏览拍摄照片。



# 测试分析（本人负责）

测试部分中除了测试整个系统的配置要求需要在多台电脑、多种环境下测试外，其他测试内容均在表5-1所示环境中测试。

表 5-1 测试环境说明表

|  |  |
| --- | --- |
|  | 环境名称 |
| 操作系统 | Windows11 |
| 视频分辨率 | 720p |
| 无人机 | Tello无人机 |
| 电脑CPU | I7-12700H |
| 电脑GPU | NVIDIA RTX3060 |

## 测试系统识别人体姿态的准确度

## 对指令识别准确性的评估需要在真实环境下进行，以将随机的干扰因素考虑在内，我们选择有人员来往的室外场地作为测试背景；此外还要考虑人体动作过程中每种动作间的连贯动作可能带来的误操作问题，因此我们随机交替进行 8 个姿态动作，每种姿态累计进行 100 次进行统计，列出姿态动作混淆矩阵如表5-2所示。为了方便在混淆矩阵上表示，用小写英文字母来指代某个人体姿态。对应关系如下：

a:双手交叉过头顶 b:双手交叉于脖子前

c:右臂张开 d:右臂内收

e:左臂张开 f:左臂内收

g:右手摸左耳 h:左手摸右耳

表 5-1 姿态测试混淆矩阵

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姿态 | A | b | c | d | e | f | g | h |
| A | 97 | 0 | 0 | 1 | 0 | 2 | 0 | 0 |
| B | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 96 | 4 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 4 | 96 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 95 | 5 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 6 | 94 | 0 | 0 |
| G | 0 | 0 | 3 | 5 | 0 | 0 | 92 | 0 |
| H | 0 | 0 | 0 | 0 | 4 | 3 | 0 | 93 |

由混淆矩阵可计算得到平均准确率为 95.375%，因为我们采用的是牺牲一定精度的MediaPipe pose模型，所以能有如此高的准确率出乎了我们的预料，不过这也反映了我们系统的优秀的可靠性和适用性。

## 测试无人机的控制功能

第一种控制方式：第一种是通过控制界面上的虚拟按钮来控制，由于是通过点击虚拟按钮来触发事件的方式来实现，所以只要代码层面没有问题，实际使用就不会有问题。经过多次测试，虚拟按钮总是能成功实现控制功能。

第二种控制方式：第二种是通过根据人体姿态对应的功能来实现对无人机的控制。在上一节测试系统识别人体姿态的准确度的过程中，无人机会对人体的姿态进行响应，此时就可以记录无人机的飞行过程和状态，以评估系统对无人机的控制功能如何。所以5.2节是在5.1节每一次动作执行之后在评估无人机的控制，所以其策略和姿态顺序是完全相同的。我们同样可以采取混淆矩阵的方式来评估。若无人机执行的是其他姿态对应的控制功能或者没有执行任何功能都算失败，正例数不会加1，只有当无人机执行的飞行功能是该姿态对应的正例数才会加1。为了方便在混淆矩阵上表示，用大写英文字母来指代某个无人机功能。对应关系如下：

A: 开启/解除距离锁定模式 B: 拍照

C: 无人机向右飞行 D: 无人机向左飞行

E: 无人机向后飞行 F: 无人机向前飞行

G: 手掌降落 H: 抛飞无人机

表 5-2 无人机控制功能测试混淆矩阵

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姿态 | A | B | C | D | E | F | G | H |
| A | 97 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 0 | 100 | 0 | 0 | 0 | 0 | 0 | 0 |
| C | 0 | 0 | 96 | 0 | 0 | 0 | 0 | 0 |
| D | 0 | 0 | 0 | 96 | 0 | 0 | 0 | 0 |
| E | 0 | 0 | 0 | 0 | 95 | 0 | 0 | 0 |
| F | 0 | 0 | 0 | 0 | 0 | 94 | 0 | 0 |
| G | 0 | 0 | 0 | 0 | 0 | 0 | 92 | 0 |
| 2 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 92 |

由图5-2可计算得到平均准确率为 762/800≈95.25%。综上所述，可见该系统无论是对人体姿态的识别准确度还是对无人机进行控制的正确率都是极高的，足以可见该系统的优秀和可靠性。

## 测试整个软件系统的配置要求

能否正常运行一个软件受制于很多因素的影响。将本系统整体打包压缩后，团队成员纷纷安装到自己以及一些同学的个人电脑上进行测试。最终得到的测试结果如表5-3所示。

表 5-3 测试环境说明表

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 操作系统 | CPU | 内存 | GPU | 系统运行平均fps |
| 个人电脑1 | Windows11 | I7-12700H | 16G | RTX3060 | 31.39 |
| 个人电脑2 | Windows10 | I5-13400F | 16G×2 | RTX4060 | 32.3 |
| 个人电脑3 | Windows11 | Ultra 5 125H | 32G | 集成显卡 | 21.49 |
| 个人电脑4 | Windows11 | Ryzen7 5800H | 16G | RTX 3050 | 28.67 |
| 个人电脑5 | Windows11 | I7-11800H | 16G×2 | RTX 3070 | 32.18 |

由于资源受限，暂无法确定出本系统所能使用的最低配置要求。但如果想运行本系统平稳持续在30fps以上，我们推荐Windows 10或更高版本的操作系统，使用的CPU频率不低于2.0 GHz，电脑应配备至少16GB的RAM，以及硬盘应至少留存300M的空间，以确保系统能够流程运行多个并行进程时不会出现性能瓶颈。

# 作品总结

## 作品特色与创新点

## 6.1.1轻量级：基于CPU，低配置，高算力。

系统基于CPU运行，实现了在低配置硬件上的高算力。这一设计突破了传统无人机系统对高性能GPU的依赖，降低了成本，同时简化了系统的部署和维护工作。轻量级设计还意味着系统对电能的消耗更低，有利于无人机的长时间飞行作业，尤其适合于农业监测、森林防火等需要长时间空中作业的场景。

### 使用了异步姿态处理实时视频流(live stream)方案：保证了实时性和高帧率。

我们系统采用了先进的异步处理机制，对实时视频流进行姿态估计，确保了系统的实时性和高帧率。该方案通过MediaPipe的Pose Landmark detection算法，实现了对人体姿态的快速识别，从而使得无人机能够实时响应用户的操控意图，提供了更为流畅和精确的控制体验。

### 高度模块化的代码结构

我们的系统代码遵循面向对象的设计原则，实现了高度的模块化和解耦。这种结构不仅提高了代码的可读性和可维护性，而且为未来的功能扩展和系统升级打下了坚实的基础。通过顶层抽象类AbsDrone的设计，系统能够兼容多种类型的无人机，增强了其通用性和可扩展性。

### 人性化的用户界面(UI)设计

系统采用了PySide2作为UI设计工具，提供了跨平台兼容性，用户界面友好，简化了操作流程。无论是初学者还是专业人士，都能快速上手。同时，欢迎界面的自动连接逻辑和操作界面的一体化控制设计，进一步提高了用户体验。

### 目标追随与姿态控制功能的创新结合

系统不仅实现了自动目标追随功能，还能够通过人体姿态来控制无人机的飞行，这一创新结合提供了更为自然和直观的人机交互方式。在农业自动化和户外运动拍摄等场景中，用户可以更便捷地操控无人机，捕捉所需画面。

### 快速部署能力

无人机和控制系统的部署简单快捷，便于迅速进入工作状态。这一特点对于紧急情况下的快速响应尤为重要，如在火灾现场的快速部署能够帮助及时收集关键信息。

### 教育意义

系统的设计和实现过程可以作为教育工具，激发学生对无人机技术的兴趣，促进STEM教育。通过教授学生无人机操作和编程，不仅能够提升学生的技术实践能力，也能够培养他们的创新思维和解决实际问题的能力。

## 应用推广

### 救火

在火灾等紧急情况下，无人机可以快速进入现场进行实时监控，为救援团队提供关键信息，提高救援效率和安全性。

1.实时监控与数据收集：在火灾现场，无人机可以快速部署，实时传输火情变化和受影响区域的高清视频，为指挥人员提供关键信息，帮助他们做出更准确的决策。

2．风险评估与资源分配：通过分析无人机传回的数据，可以评估火灾的蔓延速度和可能的高风险区域，从而更有效地分配救援资源，包括人力和灭火设备。

3.安全救援：无人机可以在不危及救援人员安全的情况下进入火场，减少救援行动中的人员伤亡风险。

### 农业

在农业领域，系统可以用于作物监测、病虫害预警和精准灌溉，推动农业生产向智能化、精准化方向发展。

作物监测：无人机可以定期飞越农田，通过高清摄像头监测作物生长状况，及时发现病虫害和营养不良问题。

1.精准农业：结合先进的图像处理技术，无人机可以用于精准农业，指导农民进行有针对性的施肥、灌溉和病虫害防治，提高农业生产效率。

2.农业灾害预防：在干旱、洪水等农业灾害高发季节，无人机可以用于实时监控农田状况，评估灾害影响，为农民提供及时的灾情信息。

3.农业教育与培训：无人机操作简便，易于编程，可以作为农业教育的工具，教授学生现代农业技术，培养未来的农业科技人才。

4.农业自动化：随着技术的进步，无人机有望实现更多自动化功能，如自动喷洒农药、自动化播种等，进一步推动农业现代化。

## 作品展望

### 集群控制

我们计划开发高效的协同算法，使多台无人机能够协同执行复杂任务。这些算法将基于分布式计算和人工智能技术，确保无人机之间能够有效通信并自主协调行动。系统将能够根据无人机的性能、电量、载荷以及任务要求，智能分配最合适的任务给每一台无人机，以提高整体作业效率和响应速度。开发实时监控和管理无人机集群的软件平台，它将提供无人机状态监控、飞行路径规划、异常干预等功能，确保集群的稳定运行。集群控制策略将包括强大的容错机制，即使在部分无人机失效或通信暂时中断的情况下，集群也能够自动调整，保持任务的连续性和系统的稳定性。在执行长期任务或面对环境变化时，无人机集群需要能够动态调整其结构，根据实时情况重新分配任务和调整飞行策略。

### 功能丰富

未来的“影随飞”系统将实现多无人机的协同作业能力，通过精心设计的算法，使得每一台无人机都能在集群中发挥最大的效能。同时，我们将持续集成更多智能化功能，如自动避障、智能路径规划等，以适应多变的应用需求。如语音控制系统。

1.自动避障：系统将持续集成自动避障功能，利用先进的感知技术，使无人机能够在复杂环境中安全飞行。

2.智能路径规划：通过机器学习算法，无人机将能够根据实时数据进行智能路径规划，提高飞行效率和安全性。

3.多模态交互：除了现有的手势控制，系统将探索更多交互模式，如语音控制、触摸屏控制等，以满足不同用户的需求。

4.数据处理与分析：系统将增强数据处理能力，不仅能够收集数据，还能进行实时分析，为用户提供决策支持。

### 商业化

随着技术的成熟，本作品将逐步推向市场，实现商业化，为更广泛的用户群体提供服务。明确“影随飞”系统的商业定位，找到最适合的商业模式。根据市场反馈，不断优化产品功能，提升用户体验，使“影随飞”系统更加贴合市场需求。

### 跨领域应用探索

探索无人机集群在不同领域的应用，如物流配送、环境监测、灾害救援等，开发领域特定的集群控制解决方案。

通过本次大赛的实践和探索，我们团队对无人机控制系统的未来发展充满信心。我们相信，通过不断的技术迭代和市场适应，“影随飞”系统将为社会带来更加积极的影响，并在无人机智能控制领域占据重要地位。

# 参考文献

1. 孔德智. 基于MediaPipe框架的人体动作识别模型在Y Balance Test中的应用研究[D].成都体育学院,2024.DOI:10.26987/d.cnki.gcdtc.2023.000330.
2. 胡昕,刘瑞安,黄玉兰.基于OpenPose改进的老人摔倒检测算法[J].现代信息科技,2023,7(23):73-78+82.DOI:10.19850/j.cnki.2096-4706.2023.23.016.
3. 张华锐,赵潇帆,王苑丞,等.基于Mediapipe的人体姿态跟随机器人系统的设计与实现[J].电脑与信息技术,2024,32(02):14-17+102.DOI:10.19414/j.cnki.1005-1228.2024.02.017.
4. 李怀义,李如晗,孙红霞.基于BlazePose模型和KNN算法的消防体能训练识别系统设计[J].人工智能,2023(03):89-97.DOI:10.16453/j.2096-5036.2023.03.009.
5. 盛洋,王健庆.基于计算机视觉的人体姿态识别研究[J].现代信息科技,2022,6(16):87-91+95.DOI:10.19850/j.cnki.2096-4706.2022.16.023.
6. Fang, H., Li, J., Tang, H., Chen, X., Zhu, H., Xiu, Y., Li, Y., & Lu, C. (2022). AlphaPose: Whole-Body Regional Multi-Person Pose Estimation and Tracking in Real-Time. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.2211.03375>
7. Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., & Grundmann, M. (2020). BlazePose: On-device Real-time Body Pose tracking. arXiv (Cornell University). <https://doi.org/10.48550/arxiv.2006.10204>
8. 高善平,李冲.基于不同PID控制器的无人机姿态控制对比研究[J].普洱学院学报,2023,39(03):24-26.
9. Cao, Z., Hidalgo, G., Simon, T., Wei, S., & Sheikh, Y. (2018). OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. arXiv (Cornell University). https://doi.org/10.48550/arxiv.1812.08008

1. 此项目在国家大学生创新训练计划平台上公布，其作品报告地址如下所示：http://gjcxcy.bjtu.edu.cn/UpLoadFileCGZB\_LW/nh1005628796995.pdf [↑](#footnote-ref-0)