

# Auto Focus Assistant

---

## 基于目标检测和立体视觉的辅助对焦系统

this is a system using a depth camera to detect objects and control a follow focus motor, aiming to drive the focus ring and achieve auto focus.

注意:本项目基于GPL3.0协议开源, 依据协议要求, 后续基于此代码的项目需保持GPL3.0开源

### background/研究背景

近年来, 随着个人Vlog和短视频的兴起, 视频拍摄的门槛越来越低, 用户对于摄影机性能的要求也越来越高  
对于个人用户或小团队用户而言, 摄影机性能主要体现在两个方向: **画质和易用性**

前者在此处不作详细展开, 后者主要包括: 体积重量、人体工学、对焦性能和防抖性能等。

其中, **自动对焦系统** 是极为重要的一部分, 而各厂家由于技术路线或专利的限制, 对焦性能参差不齐  
对于canon和sony的摄影机, 他们往往有着丰富的对焦选择和稳定的对焦性能, 得到了大量用户的认可  
而对于来自Panasonic和blackmagic design的摄影机, 他们有着出色的画质但缺乏可靠的自动对焦系统  
在当下的影像市场上, 有很多画质优秀的影像设备由于易用性不佳, 而导致了大量用户的流失

- 本项目旨在改善摄影机的自动对焦性能, 基于目标检测和追踪, 实现自动或半自动对焦

### principle/主要原理

- 本项目属于主动式对焦系统  
基于结构光立体视觉技术, 通过深度相机获取深度图和彩色图, 并将二者进行匹配  
通过计算机视觉技术检测并追踪图像中的目标, 包括人脸、人体和其他物体, 并实时读取目标的距离  
通过距离-电机位置映射函数, 驱动电机带动对焦环至目标位置, 完成对焦动作

### hardware structure/硬件系统结构

- **range finder/测距仪**
  - 用于测量目标与相机的距离, 常用的测距仪包括深度相机和各类激光雷达
  - 本项目所使用的测距仪是两款深度相机: 奥比中光AstraPro 和 英特尔Realsense D415(兼容435/455/457, 未来可能部分兼容D430)
  - 前者的原理的单目红外结构光测距, 在室内表现优异, 有效距离8m; 后者为双目结构光, 在室内外均可使用, 理论有效距离>10m
  - 由于个人精力有限, 大部分工作将基于Realsense相机开发, 不能保证对AstraPro提供持续的兼容
- **processor/处理器**
  - 用于处理从测距仪得到的数据, 常用的视觉处理平台有英伟达的NX系列、树梅派等平台 and 以NUC为代表的小型电脑
  - 综合考虑成本、体积重量和集成化, 本方案所采用的处理器之一为英特尔的Computer Stick(电脑棒), 它采用X86架构, 能够方便地进行程序开发和运行
  - 另一套方案是本项目当前采取的方案: 即采用分布式方案, 将算力平台和相机分离, 以牺牲集成度的方式大幅降低了成本和重量, 并且不需要妥协性能

- **operator/执行器**

- 执行器是直接驱动镜头进行对焦的组件，一般分为内置的对焦马达和外置的跟焦电机
- 目前所使用的执行器为外置42闭环步进电机，后续可能会更换为28步进电机或无刷电机(V2.0版本中更新为铁头公司的原力N跟焦马达)
- 同时，基于ESP32进行蓝牙连接部分相机（如BMPCC和ZCAM）的研发也在进行，便于驱动内置马达的自动镜头

- **controller/控制器**

- 控制器用于实现“手自一体”的自动对焦模式选择和功能切换
- Tilta铁头公司是一家专业制作相机配件的器材供应商，他们于2018年底推出了原力N(NucleusN)电动跟焦系统  
这套系统在这几年间不断更新壮大，提供了跟焦马达、跟焦手柄和蓝牙跟焦手轮等一系列解决方案  
本项目也采用了铁头公司的原力N系统，主要针对其跟焦马达和跟焦手柄进行研发
- 在本项目中，通过逆向功能还原其通信协议，原力N系统的跟焦手柄被改造为控制手柄  
本项目对按键和滚轮等功能进行了重映射，具体的使用方法见后文“控制逻辑”部分  
除此之外，该手柄还承担给执行器电机供电的功能

- **monitor/监视器**

- 监视器是监控对焦模式和状态、展示对焦波形图的关键部件
- 本项目从v2.0版本开始加入了监视器，兼容任意摄影监视器(有hdmi输入即可)
- 本项目目前采用opencv自带的窗口显示函数进行画面输出，暂未搭建完善的图形界面，因此也不支持触摸操控
- 监视器画面左侧展示对焦参数，右侧展示对焦波形图，波形图中用红线展示展示滚轮编码器位置，用绿线展示电机实时对焦位置

## software structure/代码文件结构

### up/上层

- **main**

- main函数：程序入口

- **controller/控制器**

- 控制程序运行,负责切换校准模式和对焦模式

- **SubController/子控制器**

- 以下两个控制器的基类，未来可能兼容更多的模式

- **CalController/校准模式控制器**

- 在校准模式下，调用控制各个modules

- **FocusController/对焦模式控制器**

- 在对焦模式下，调用控制各个modules

## **core/核心层**

- **modules/核心模块:分为以下子模块**

- **calibrator/校准器**

- 负责对镜头进行校准，建立查找表并写入lens\_param.yml

- **detector/识别器**

- 负责识别目标
    - FaceDetector:基于卷积神经网络(Yunet)的面部识别，已部署
    - FaceDetectorLight:基于SCRFD的面部识别，已部署
    - ObjectDetector:基于YOLOv5s的目标检测网络，已部署但无法实时检测，暂时搁置
    - ObjectDetectorLight:基于YOLO\_fastestV2的目标检测网络，已部署

- **dis/距离获取器**

- 负责距离的解算和滤波

- **motor/电机驱动器**

- 负责与电机间通信
    - 第一代AF\_assistant:基于42闭环步进电机
    - 第二代AF\_assistant:基于铁头NucleusN电机

- **reader/读取器**

- 负责从深度相机读取彩色视频流和深度信息流
    - 第一代AF\_assistant:基于奥比中光相机
    - 第二代AF\_assistant:基于Realsense相机

## **bottom/底层**

- **serial/通信器**

- 用于和电机/相机/单片机等进行串口通信，目前仅支持电机通信
  - Data类为总协议，分为以下子通信协议：
    - TransferData: 程序内部传递数据汇总
    - SendData1/2/3/4/5/6: 程序向电机发送的指令
      - 1~4分别对应步进电机的触发指令、读取指令、修改指令、运动指令
      - 5~6分别对应NucleusN电机的运动指令、校准指令
    - Read:
      - 对于步进电机:程序从步进电机读取当前脉冲值
      - 对于NucleusN:程序从手柄读取数据

- **param/参数文件**

- 用于注入参数，目前分为以下两类参数：
  - param: 普通参数，为定值，运行前可以调整
  - lens\_param: 镜头标定参数，储存各个镜头的查找表数据，运行时可能改变（如校准/新建镜头）

## **user instruction/使用方法**

### **install dependent lib/安装依赖库**

- OpenCV 4.5.0或更新版本(推荐4.5.4及以上)
- Realsense SDK 2.0

### **run/运行程序**

- 编译代码 (在项目文件夹下)

```
mkdir build
cd build
cmake ..
make
```

- 执行程序 若编译成功，则可以进一步执行程序

```
./AF_assistant
```

运行程序后，会提示是否找到原力N

如果出现以下提示:

找不到原力N

请检查接线，并重新启动程序

若能够找到原力N，则可以进一步选择是否执行自动校准(换镜头后必须校准，其他随意)

即将执行自动校准, 输入1确认, 输入2跳过:

(理论上可以支持无限位镜头手动校准，后续有空就加)

校准完成或跳过后，可以选择镜头型号（1~5）,如果需要新增镜头，则按0

请输入镜头编号(1/2/3/4/5) 若新建镜头请按0 回车确认

以直接执行对焦功能为例，输入镜头编号(如2),回车即系统开始运行  
如果需要开机自启动，可以在.yml文件中设置好上述参数，即可直接运行

### auto start/设置开机自启动

由于本项目需要用到imshow，所以开机自启动的设置会略为复杂，也希望有大佬提出改进建议  
若采用传统的systemd去启动程序，由于imshow的存在会报错  
故本项目自启动思路如下:将项目做成快捷方式，再用系统自带的开机自启动去启动快捷方式  
操作步骤如下

- 在 /usr/local/下创建一个文件夹，名为af-assistant
- 进入该文件夹，在文件夹内部创建两个.sh文件，分别命名为af-start.sh和gtk-launch.sh(权限问题不再赘述)
- 前者用于进入项目文件夹并启动项目代码，后者用于进入快捷方式文件夹并启动快捷方式
- 前者内容示例如下(可根据代码位置修改)

```
#!/bin/bash

cd /home/alex/AF_assistant/auto-focus-assistant/build
./AF_assistant
```

- 后者内容示例如下(可根据代码位置修改)

```
#!/bin/bash

cd /usr/share/applications/
gtk-launch af-start.desktop
```

- 随后进入/usr/share/applications/,在该文件夹下创建文件，名为 af-start.desktop
- 内容如下:

```
[Desktop Entry]
Version=2.0
Type=Application
Name=AF_assistant
Exec=/usr/local/af-assistant/af_start.sh
Terminal=true
Comment=execute af_start.sh
```

- 最后在桌面应用中，找到名为"启动应用程序"的应用，打开
- 点击添加，名称可以随意填写，命令填写gtk-launch.sh的地址(/usr/local/af-assistant/gtk-launch.sh)

### power limit/功率限制

操作步骤如下

- 在/usr/local/下创建一个文件夹，名为powerlimit
- 进入该文件夹，创建名为powersave-mode的.sh文件
- 文件内容如下(需要对应具体线程数量进行修改!)(以下以8c16t的4800U处理器为例):

```
#!/bin/sh

cpufreq-set -g powersave -c 0
cpufreq-set -g powersave -c 1
cpufreq-set -g powersave -c 2
cpufreq-set -g powersave -c 3
cpufreq-set -g powersave -c 4
cpufreq-set -g powersave -c 5
cpufreq-set -g powersave -c 6
cpufreq-set -g powersave -c 7
cpufreq-set -g powersave -c 8
cpufreq-set -g powersave -c 9
cpufreq-set -g powersave -c 10
cpufreq-set -g powersave -c 11
cpufreq-set -g powersave -c 12
cpufreq-set -g powersave -c 13
cpufreq-set -g powersave -c 14
cpufreq-set -g powersave -c 15

exit 0
```

自启动方面:

- 方便起见，可以直接在上述的af-start.desktop中进行修改
- 在文件开头加入以下两行内容即可（修改为用户密码）

```
cd /usr/local/powerlimit/
echo "你的密码"|sudo -S ./powersave-mode.sh
```

## control logic/控制逻辑

- NucleusN手柄控制逻辑 & 决策器控制逻辑 按键重映射逻辑如下:
- 开机和关机 单击开机键开机，长按开机键关机(未改变原始设定)
- 总模式切换 总模式切换可以在开机后单按REC键(同开机键)实现，同时手柄指示灯颜色改变  
总体分为两大模式：自动对焦(绿灯，AF)和手动对焦(红灯，MF)  
注意，当模式由自动对焦切换至手动对焦时，默认为锁定状态(用于规避临时障碍物)  
只有手动转动滚轮，才能进入全手动模式
- 自动对焦
  - 滚轮:用于目标切换  
滚轮用于实时选择不同的目标

选择逻辑如下:

滚轮对应编码器数值0-9999, 通过非线性重映射到实际距离

通过滚动滚轮可以切换远近目标

当滚轮放在最左端, 默认对焦最近的目标, 且不进行追踪

当滚轮放在最右端, 默认对焦至背景(待实现)

- CAL按键(待实现):用于自动对焦下模式选择

CAL按键可以切换自动对焦模式, 按下按键用于循环切换三种模式

基于神经网络的人体&人脸混合智能对焦

中心点对焦

中心区域对焦

未来将开放更多的物体识别模式

- 手动对焦

- 滚轮:用途和直接驱动原力N电机一致

- CAL按键:用于常规模式和精准模式切换

常规模式即将编码器0-9999线性映射至电机读数的0-999

精准模式即缩小编码器的映射范围

- 附加功能:

自动对焦/手动对焦下长按CAL按键:暂停系统运行

之后按下REC按键:解除暂停, 系统恢复运行

## test platform/测试平台

- CPU: ryzen7 4800u
- 系统: ubuntu 20.04
- 电池容量: 损耗后剩余46wh
- 测试环境: 关闭wifi和蓝牙, 内屏最低亮度, 视野内持续追踪单一目标, 所有核心限制在最低频率 (1.4Ghz)
- 测试结果: cpu核心平均占用约为40%, 内存平均占用3.15G(空闲时为3.05G) 平均帧率维持在17fps上下, 较为流畅 续航时间(99% - 5%)约为2h20min(推算, 可能会有浮动) 以此类推, 采用95wh希铁外接电池或99whV口电池可以供电4.5-5h 采用77wh的F970电池或75wh的希铁外接电池可以供电3.5-4h

## BOM/物料清单

- **essential components/必备部件**

- **摄影配件**

- 铁头 原力N电机 / Tilta NucleusN motor
    - 15mm 碳纤维导管 / 15mm carbon fiber tube
    - 15mm 单孔管夹 / 15mm pipe clamp
    - 摄影多孔芝士板 / photography cheese plate
    - 1/4 螺丝 / 1/4 screw

- **其他配件**

- 英特尔 D415/D435深度相机 / Intel Realsense D415/D435 depth camera
    - MSCF 滑移支架(固定相机) / MSCF drift support (camera support)

- cforce拓展坞 / cforce typec dock
- 相机信号线(usb3.0 typec-a) / camera cable (usb3.0 typec-a)
- 定制控制手柄信号线 / DIY control handunit cable
- 全功能typec 主机通信线(长) / USB4 typec (long)
- 3M 双面胶 / 3M double-side tape

- **optional components/选配部件**

- **摄影配件**

- HDMI 监视器 / HDMI monitor
    - 监视器支架 / monitor support
    - HDMI 公对公线 / HDMI cable
    - 铁头原力N 控制手柄 / Tilta NucleusN hand unit
    - 控制手柄连接件 / hand unit adapter

## **important updates/重要更新**

- 2023.5.6 修复此前测试中的bug  
修改reprojection函数逻辑, 性能大幅提升  
新增基于SCRFD的面部识别, 速度和Yunet基本一致但侧脸识别率大幅提升
- 2023.5.3 完成首次室外测试  
发现了两个隐藏BUG, 正在尝试修复  
总体测试效果还行, 大部分功能正常  
后续方向:修复bug, 完善手柄策略(尤其是解决按键识别率的问题)  
给面部识别也加上strict-in策略  
其他问题:散热问题较为严重, 持续运行超过五分钟机身温度就会非常烫手  
(室外温度接近30度, 电脑放置于斜挎包内部)
- 2023.4.30 四月份最后一次更新  
完整demo已经上线, 新增电机响应控制、ROI选取和控制手柄策略(部分实现)  
只需要一个USB-TTL, 拆分为两个半双工模式即可
- 2023.4.25 NucleusN接收和发送数据测试成功, 但是:  
同时接收手柄数据并发送电机数据需要两个USB-TTL模块  
意味着需要同时打开两个串口, 需要预先绑定串口
- 2023.4.20 基本完成决策器和tracker的重写(主要是tracker)  
兼容YOLO\_facstestv2,具体逻辑待完善
- 2023.4.09 放弃nuitrack  
效果差&SDK烂&还一堆限制  
部署 YOLOv5s & YOLO\_fastestv2,以后者为主, 前者无显卡很难达到实时  
准备改进决策系统
- 2023.4.06 部署nuitrack sdk  
这玩意真的过于阴间, 此处记录一下安装步骤
  - stage 1: 下载和安装



- 首先在github上找到nuitrack\_sdk，下载Platforms文件夹中对应的安装包（ubuntu amd64），打开并安装
- 用第三方工具下载其Example和Nuitrack文件夹(完整下载SDK体积会很大)
- stage 2: 激活和测试
  - 接下来请登陆Nuitrack官网，申请trial license（试用版本），会收到相应包含激活码的邮件
  - 此时打开ubuntu命令行，输入nuitrack，会出现图形界面
  - 随后插入深度相机，激活（会和相机固件绑定）并测试
- stage 3: 部署必要文件
  - 将Nuitrack文件夹复制/剪切到和Auto Focus Assistant文件夹所在位置相同的位置（作为AF assistant的子文件夹）
  - 将Examples中的multisensor文件夹打开，将它提供的opencv3.4headers解压到Nuitrack文件夹的include目录中
    - （此时include文件夹中应含有opencv子文件夹和opencv2子文件夹）
- 2023.3.15 基本完成顶层模块的重构和函数优化  
下一步将继续优化modules，尤其是新增的tools工具库，以及完善奥比中光相机的reader
- 2023.3.12 取消processor,将原有的processor模块分散至dis、motor、detector、reader、tools等模块  
将原FrameProcessor改为FocusController,从原Calibrator分出CalController,加入顶层结构