

叠动科技软件使用手册

版本： 0.3.0

更新日期： 2026 年 3 月 6 日

简介

本文件介绍如何上手和运行叠动视触觉传感器配套的开发套件。本套件是配合叠动视触觉传感器进行图像处理，计算形变、力以及接触信息。它可以从摄像头或视频文件中读取图像，处理后打印和展示处理后信息。

支持操作系统：Windows、Ubuntu

建议 Python 版本: 3.10

基础环境设置

安装 Python 与 Anaconda

1. 访问 [Anaconda 官网](#) 下载适合您操作系统的安装包（Windows、Linux）。
2. 按照提示完成安装。（Windows：建议在安装过程中勾选“将 Anaconda 添加到系统环境变量”选项，方便后续使用）。
3. 安装完成后，打开命令行终端（Windows 用户可在“Anaconda Navigator”打开“Anaconda Prompt”），输入 `conda info`，若能显示 `conda` 的信息，则说明安装成功。

创建 conda 虚拟环境

虚拟环境相当于一个独立的 Python 工作空间，可以避免不同项目所需的库版本冲突。打开命令行终端（或 Anaconda Prompt），依次输入以下命令(以 python 3.10 为例)：

```
conda create -n orisys python=3.10
conda activate orisys
```

激活后，命令行提示符前会出现 (orisys) 字样，表示当前已进入虚拟环境。

创建工作区文件夹

“Anaconda Prompt” 启动后的默认工作路径在用户根文件夹中，建议创建工作区文件夹方便管理，可以手动创建或者运行 `mkdir <path>` 命令：，例如

```
mkdir orisys_ws
```

创建后如果要更换工作区，使用 `cd <path>` 命令，例如：

```
cd "C:\Users\orisys\orisys_ws"
```

安装 CUDA 和 cuDNN（可选）

如果您拥有 NVIDIA 显卡，并希望利用 GPU 加速计算，则需要安装 CUDA 工具包和 cuDNN。

1. 方法一（官网下载）：访问 [NVIDIA CUDA Toolkit 官网](#) 下载对应系统的 CUDA Toolkit 安装包，并按照提示安装，注意 12.X 和 13.X 的版本对应不同的 CuPy 库版本。
2. 方法二（使用 Anaconda 安装，推荐）：在已激活的 orisys 环境中直接运行以下命令（以 CUDA 12.9 为例）：`conda install nvidia/label/cuda-12.9.0::cuda-toolkit nvidia::cudnn` 该方法会自动配置好 CUDA 和 cuDNN，无需手动下载。

注意：如果您的计算机没有 NVIDIA 显卡，或者不想使用 GPU 加速，可以跳过此步骤。套件也支持纯 CPU 运行（但速度可能较慢）。

安装 orisys SDK

orisys SDK 以 .whl 文件形式提供，您需要根据操作系统、Python 版本和硬件平台选择正确的文件。

下载正确的 .whl 文件

从官方渠道获取 .whl 文件，文件名格式通常为： orisys-{版本}-{Python 版本}-{Python 版本}-{操作系统}_{硬件架构}.whl 例如： orisys-0.3.0-cp310-cp310-win_amd64.whl 表示：

版本： 0.3.0 Python 版本： 3.10（cp310） 操作系统： Windows（win） 硬件架构： 64 位（amd64）

请根据您的系统选择对应的文件，或者咨询客服索取：

安装 .whl 文件

将下载好的 .whl 文件放在一个文件夹中（如上述 orisys_ws 文件夹），打开命令行（确保已激活 orisys 环境或使用“ Anaconda Prompt” ），使用 cd 命令切换到该文件夹，然后运行：

```
# 请将文件名替换为您实际下载的文件名  
pip install orisys-0.3.0-cp310-cp310-win_amd64.whl
```

安装 CuPy（可选）

注意：如果您没有安装 CUDA，或者使用 CPU 运行，可以跳过这一步。

CuPy 是一个与 NumPy 类似但支持 GPU 加速的库，您需要根据您安装的 CUDA 版本选择对应的 CuPy 版本。

方法一（使用 extras 参数，推荐）： 在安装 orisys 时，可以直接指定 CUDA 版本，pip 会自动安装配套的 CuPy：

```
# 如果您的 CUDA 版本是 12.x  
pip install orisys[cuda12]  
  
# 如果您的 CUDA 版本是 13.x  
pip install orisys[cuda13]
```

如果之前已经安装了 orisys，可以单独安装 CuPy：

方法二（手动安装）：

```
# CUDA 12.x
pip install cupy-cuda12x>=12.3.0

# CUDA 13.x
pip install cupy-cuda13x>=13.0.0
```

测试安装

安装完成后，在命令行输入“python”命令进入 Python 环境，并导入 orisys 库并查看版本号，验证是否成功：

```
python
import orisys
print(orisys.__version__)
```

如果输出版本号（例如 0.3.0），则说明安装成功。

使用方法

运行示例程序

套件里面包含了例子展示用法，运行步骤如下：

Windows:

1. 检查是否有其他正在连接的 USB 摄像头设备：打开 powershell, 运行命令 `Get-PnpDevice -PresentOnly | Where-Object { $_.Class -match '^Camera' }`，如果没有的话不会有输出。
2. 把传感器的 USB 头与电脑连接，再次运行步骤一的命令，会新增一行输出

3. 如果第一步没有输出，传感器的 ID 就是 “0” ，如果第一步有一个设备，传感器的 ID 就是 “1” ，如此类推。
4. 在确定了传感器的 ID（下方例子假设为 “0” ）后，打开 Anaconda Prompt, 切换到工作路径，然后在命令行中输入：

```
python example1_basic_pipeline.py -v 0
```
5. 程序会运行，并自动打开可视化窗口和打印资料，在窗口上按键盘 “q” 键可以停止运行，按 “r” 键可以重设数据

Ubuntu:

1. 检查是否有其他正在连接的 USB 摄像头设备：打开终端, 运行命令 `v4l2 -list-devices`，如果没有的话不会有输出。
2. 把传感器的 USB 头与电脑连接，再次运行步骤一的命令，会新增输出
3. 根据新增的输出，例如 “ /dev/video0” ，确定传感器的 ID 为 “0” ，如果是 “ /dev/video3” 的话 ID 为 “3” ，如此类推。
4. 在确定了传感器的 ID（下方例子假设为 “0” ）后，打开终端, 激活相应的 python 环境，切换到工作路径，然后在命令行中输入：

```
python example1_basic_pipeline.py -v 0
```

5. 程序会运行，并自动打开可视化窗口和打印资料，在窗口上按键盘 “q” 键可以停止运行，按 “r” 键可以重设数据

开发套件使用方法

参考开发套件附带的开发文档。