

1、实验名称及目的

相机标定实验：图像测量过程以及计算器视觉中，为确定空间物体某点的三维几何关系位置与其在图像中对应点之间的相互关系，必须建立相机成像的几何模型，模型的参数就是相机的参数。

2、实验原理

首先通过 vis.jsonLoad() 加载传感器配置文件，其参数配置如下

“SeqID”代表第几个传感器。在本例程中为单目相机标定所以只选用了一个相机，此处 0 代表第一个相机。

“TypeID”代表传感器类型 ID，1:RGB 图（免费版只支持 RGB 图），2:深度图，3:灰度图。

“TargetCopter”传感器装载的目标飞机的 ID，可改变。

“TargetMountType”代表坐标类型，0: 固定飞机上（相对几何中心），1: 固定飞机上（相对底部中心），2: 固定地面上（监控）也可变。

“DataWidth”为数据或图像宽度此处为 640，“DataHeight”为数据或图像高度此处为 480。

“DataCheckFreq”检查数据更新频率此处为 30HZ。

“SendProtocol[8]”为传输方式与地址， SendProtocol[0] 取值 0: 共享内存（免费版只支持共享内存），1: UDP 直传 png 压缩，2: UDP 直传图片不压缩，3: UDP 直传 jpg 压缩； SendProtocol[1-4] : IP 地址； SendProtocol[5] 端口号。

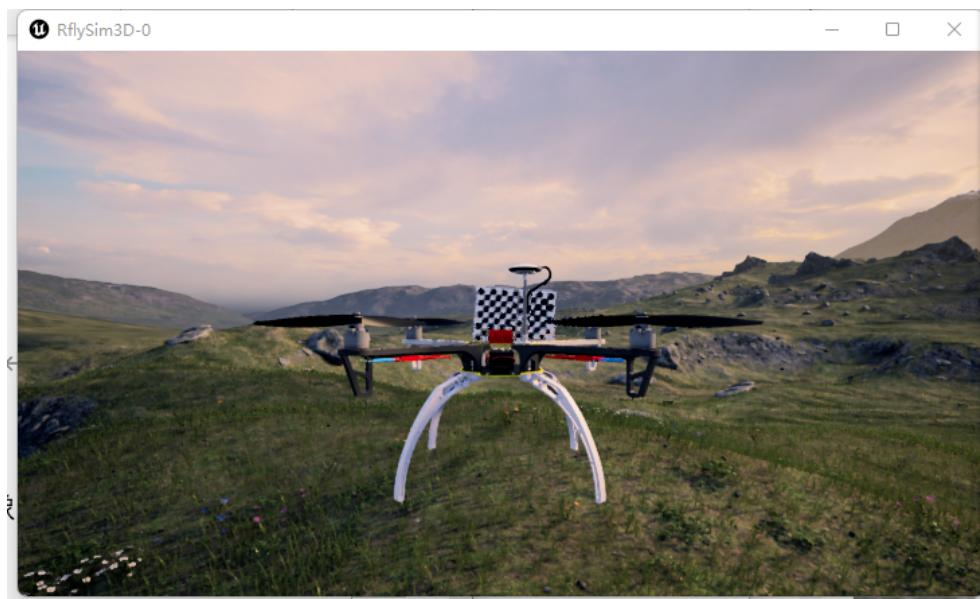
“CameraFOV”为相机视场角（仅限视觉类传感器），单位度也可改变。

“SensorPosXYZ[3]”为传感器安装位置，单位米也可改变。

“SensorAngEular[3]”为传感器安装角度，单位度。也可改变。

将传感器的图像通过死循环显示在屏幕上，并保存在一个以当前日期和时间命名的文件夹中，最后通过 MATLAB 进行标定。

3、实验效果





4、文件目录

文件夹/文件名称	说明
20220207_220418	捕获的图片。
OneCameraCal.bat	一键启动脚本。
OneCameraCal.py	单目相机标定例程。
bord-Calibration.py	内参矩阵。
ball2-Calibration.py	双小球标定例程。

5、运行环境

序号	软件要求	硬件要求	
		名称	数量
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台免费版		
3	Visual Studio Code		

①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

6、实验步骤

Step 1:

右键点击 3-VisionAI API\2.CameraCalcDemo\OneCameraCal.bat 一键启动脚本，选择以管理员身份运行。

CalibrationPrinciple.docx	2022/10/30 0:12	Microsoft Word ...	5,401 KB
CalibrationPrinciple.pdf	2023/10/17 16:19	Microsoft Edge ...	1,720 KB
Config.json	2022/6/11 23:40	JSON File	1 KB
OneCameraCal.bat	2022/9/20 17:07	Windows 批处理...	1 KB
OneCameraCal.py	2022/3/21 20:47	Python File	4 KB

会自动打开 RflySim3D 仿真平台。

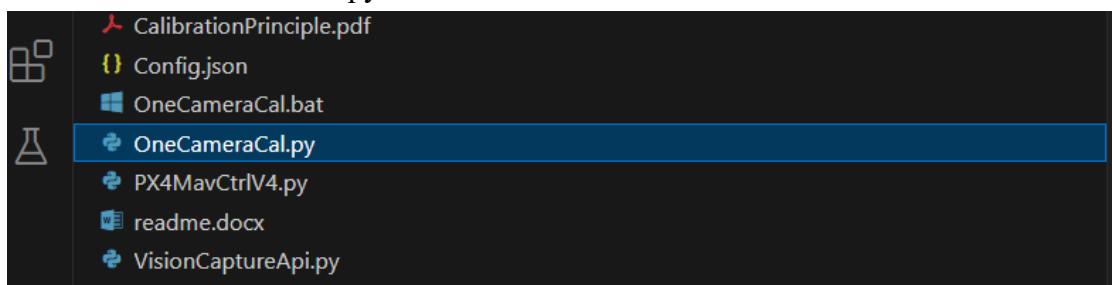


Step 2:

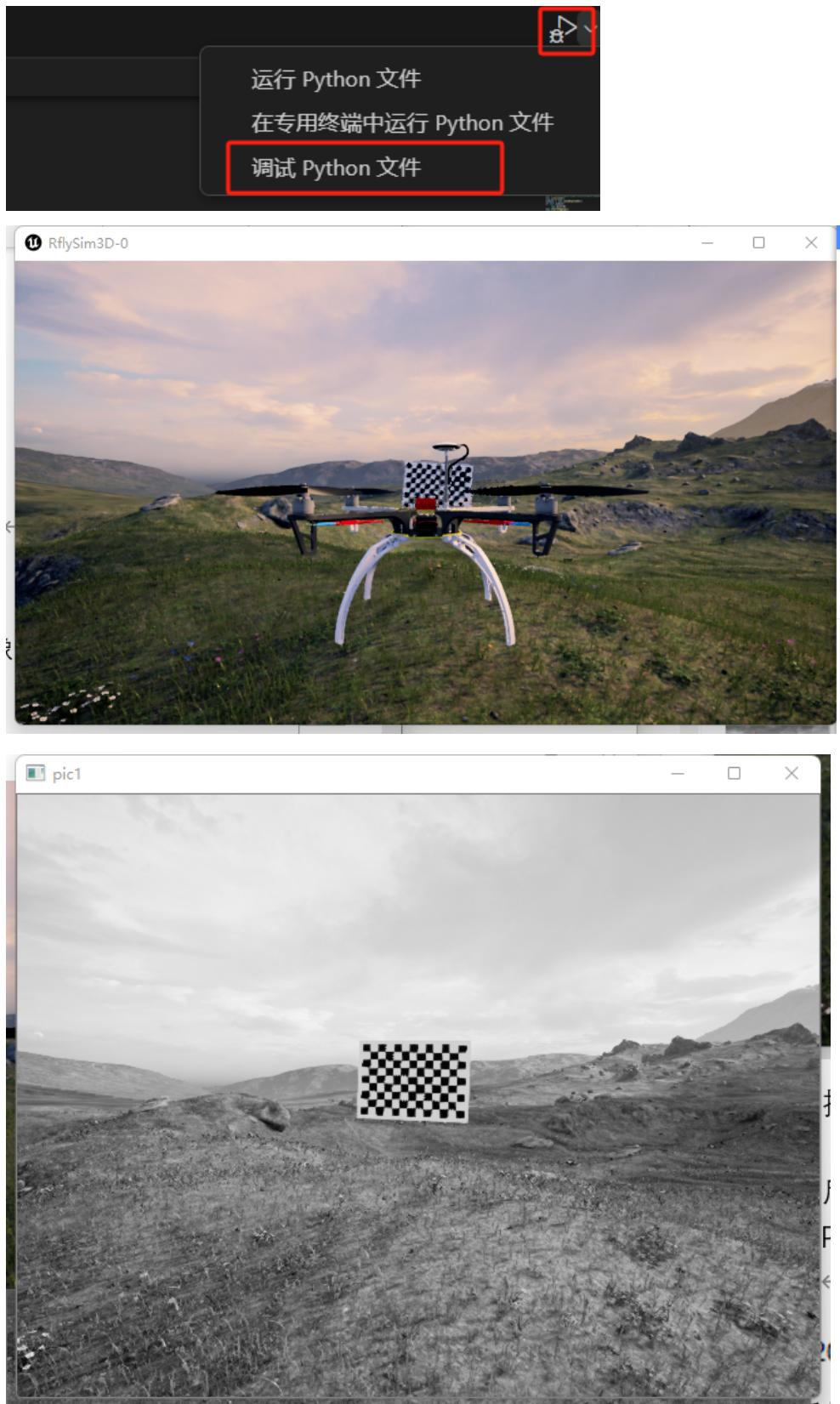
通过 Visual Studio Code 打开 3-VisionAI\3.CameraCalcDemo 文件夹。



点击 OneCameraCal.py 文件。



点击运行此文件。



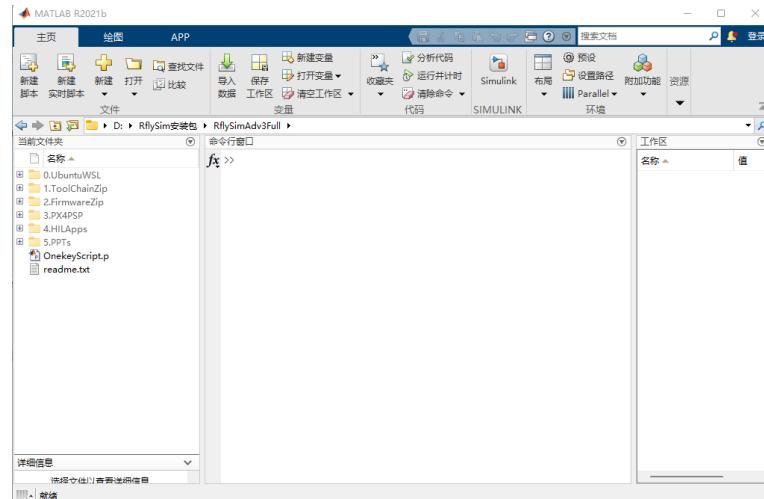
若标定板与相机的距离太远或太近，会影响检测结果，可以修改 InitTargePos 中的第一个数值来改变距离，获得三十张图片左右后就可将程序停止。

Step 3:

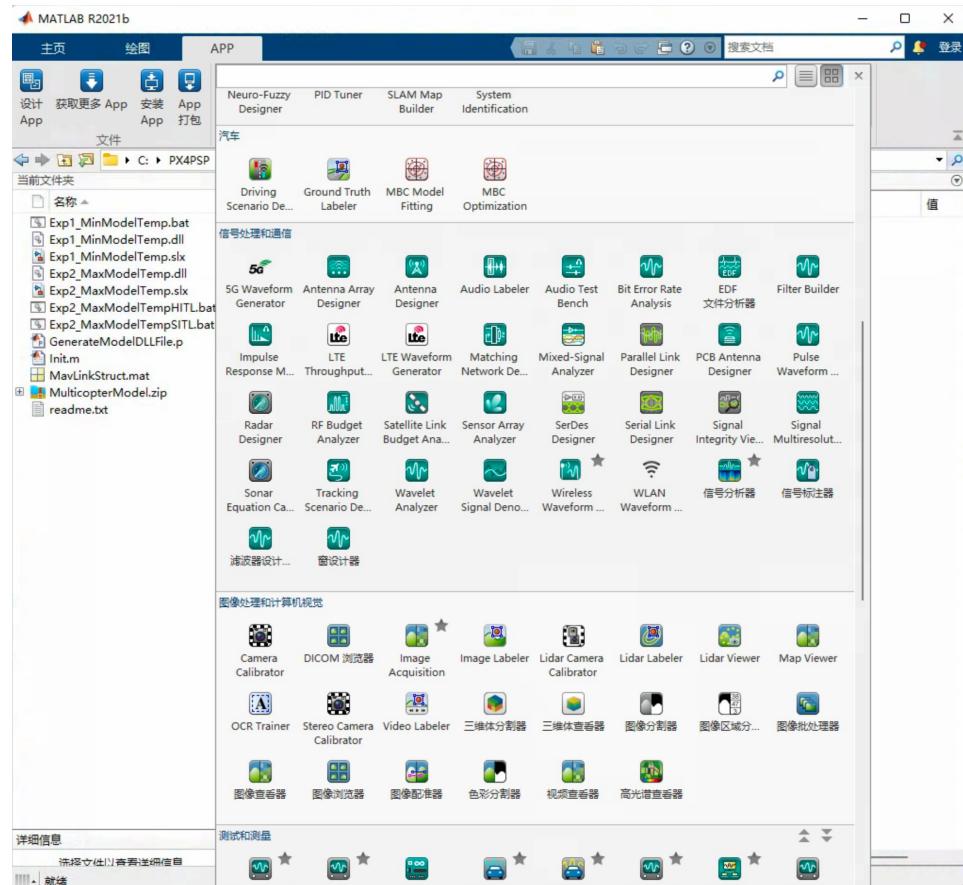
运行 PX4PSP\ReLabPath.py 文件。

Step 4:

将存储下来的单目相机图像，在 MATLAB 中校准，得到相机内参。
打开 matlab，点击上方的 APP 栏。

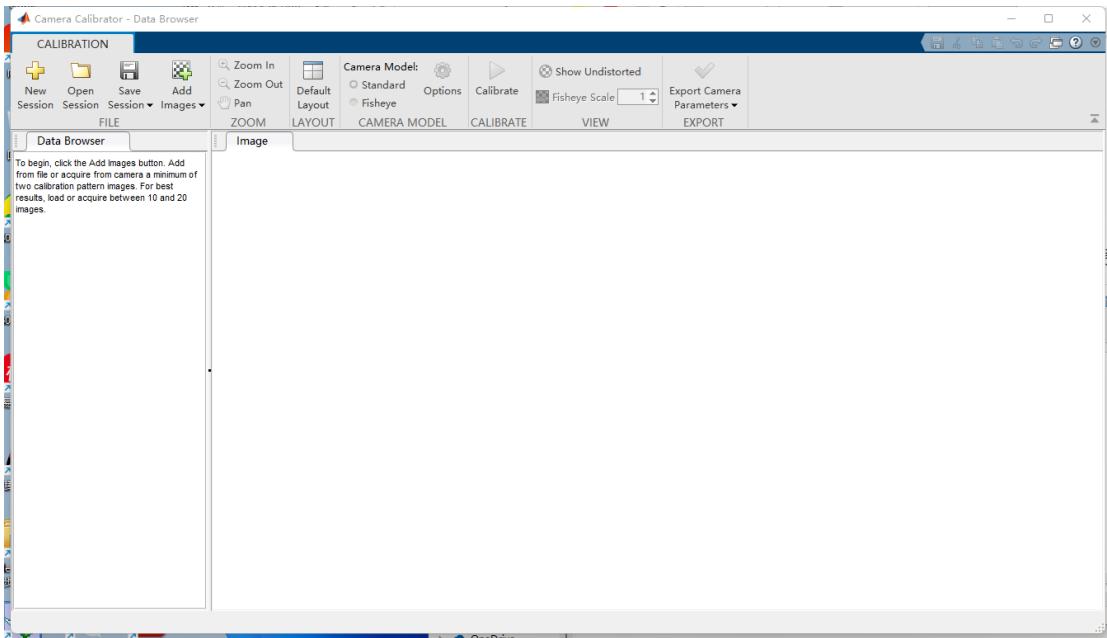


下拉工具栏，选择 Camera Calibrator 工具箱

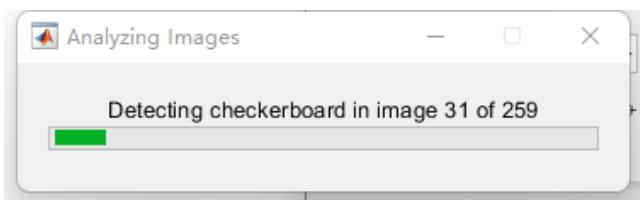
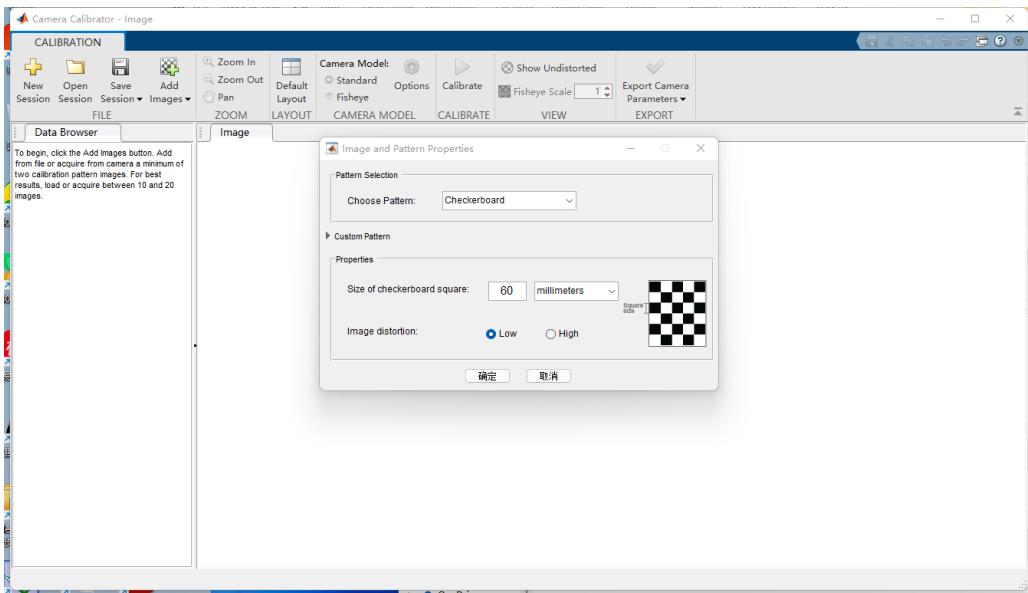


打开工具箱后，点击 Add Images，打开到我们抓取图片的位置，选取图片。选取完成后

点击打开按钮。



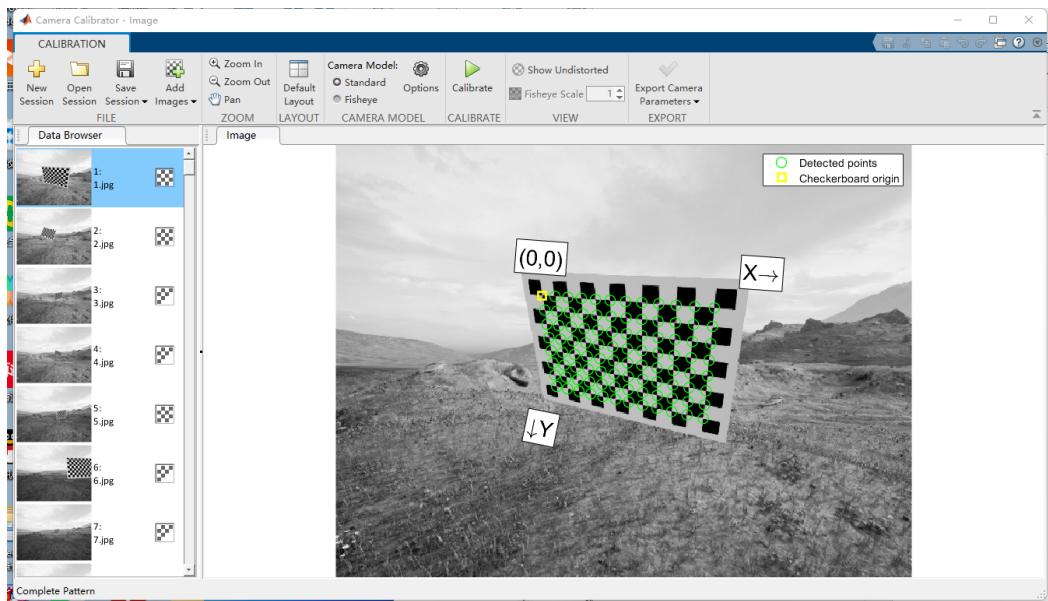
弹出下面窗口，输入棋盘格的大小。场景中为大家提供的是 60mm，点击确定。



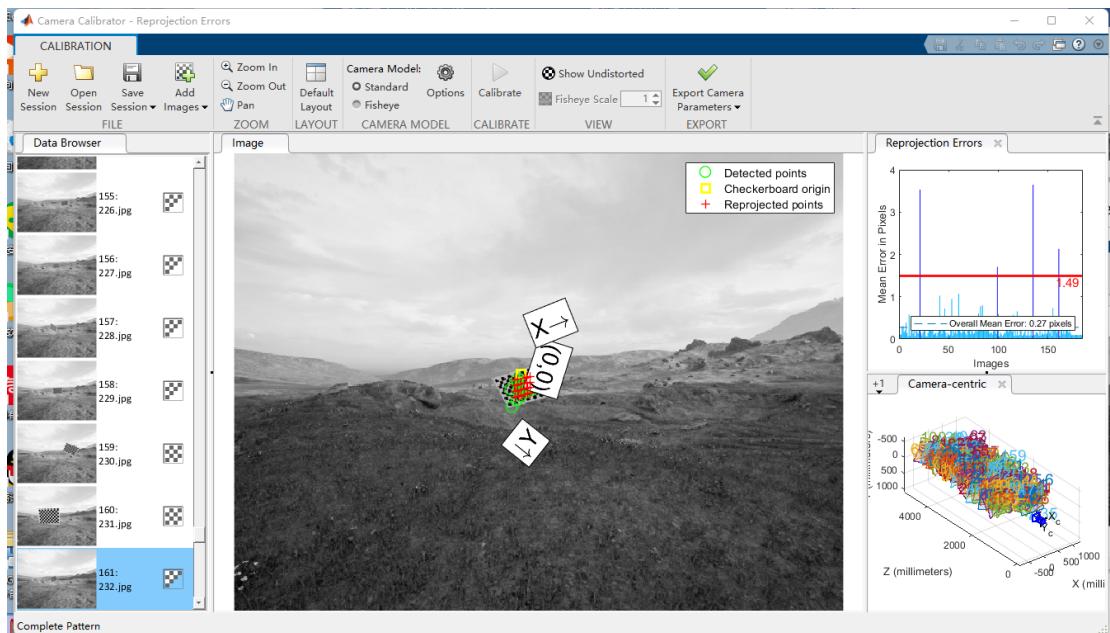
几秒钟后，工具箱会告诉你接受的图片，直接点确定。



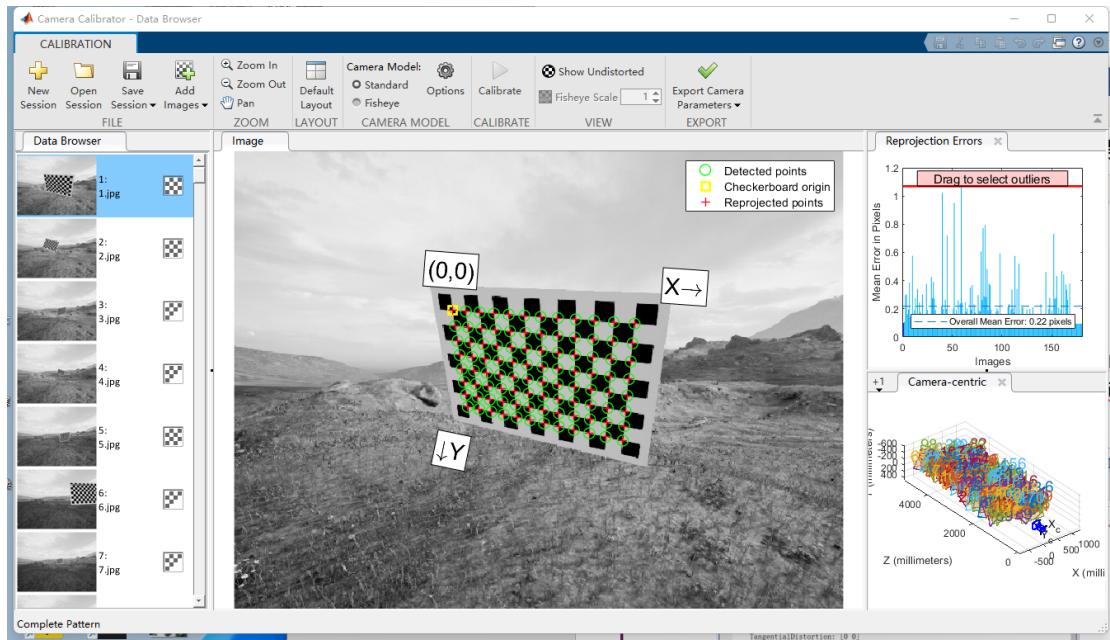
可以大概浏览一下棋盘格提取质量，应该是都在图像上。点击 Calibrate 按钮开始标定。



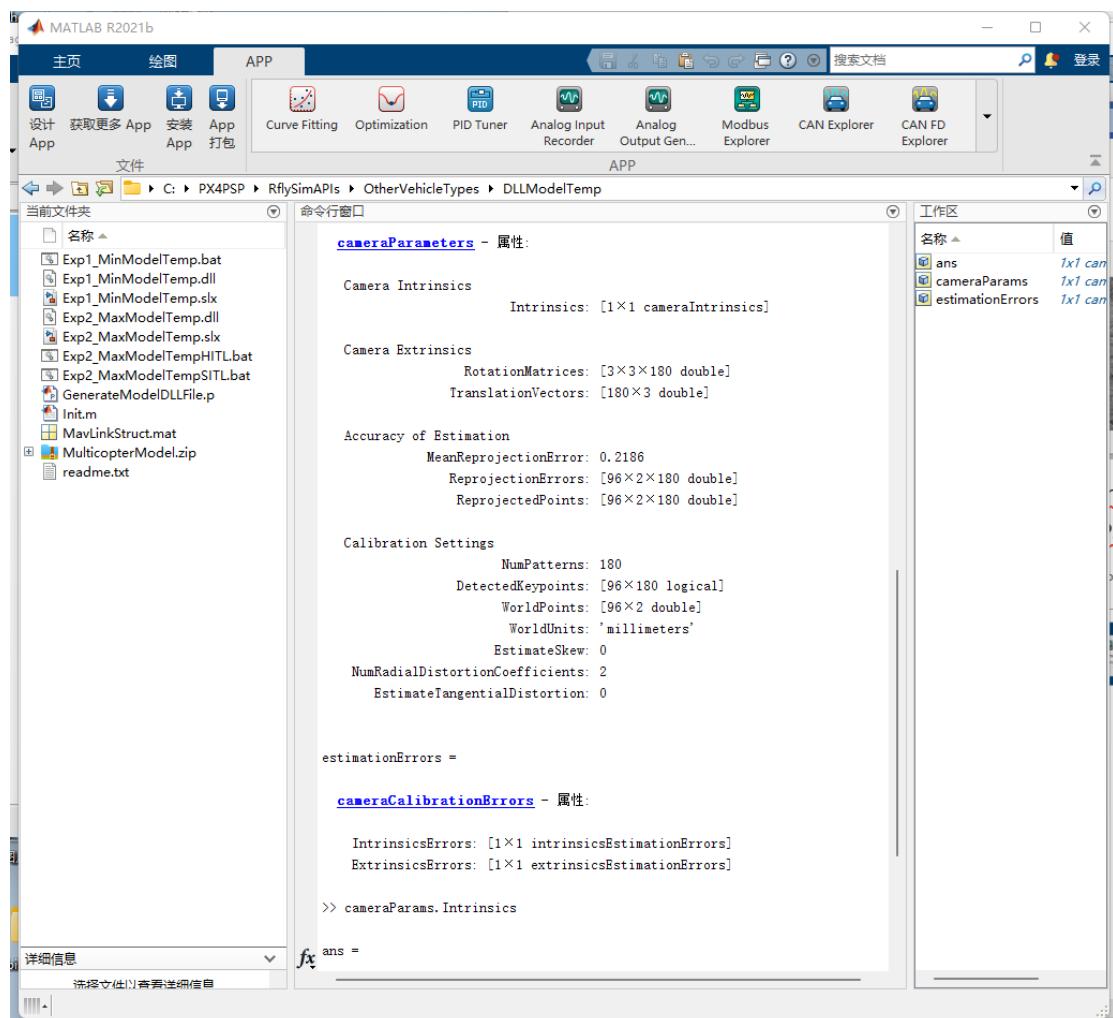
可以拖动红线选择反投影误差较大的图片，按 Delete 删除这些图片并重新标定。



结果满意之后点击 Export 导出标定的结果。



回到 Matlab 工作区，会看到一个名为 cameraParams 的结构体，双击可以看到各成员变量。也可以在下方输入 cameraParams.Intrinsics 查看相机信息，输入 cameraParams.IntrinsicMatrix 查看内参矩阵。



Step 5:

在 bord-Calibration.py 文件中将第 19 行的路径改为要标定的照片路径并运行。

```
#标定图片路径  
images = glob("./20220207_220418/img1/*.jpg")  
for
```

Step 6:

点击 ball2-Calibration.py 文件。



点击运行此文件。



7、参考文献

[1] 无

8、常见问题

Q1: 无

A1: 无