

培训文档：使用 MATLAB 标定相机

版本：V1.0

编写日期：2025-09-16

编写人：张俊杰

适用对象：26 赛季视觉组梯队

目录

1. 前言	3
2. 标定原理简述	3
3. 实验准备	3
4. 图像采集	4
5. 使用 MATLAB 标定工具箱	5
5.1 打开标定工具	5
5.2 导入图像	6
5.3 自动检测角点	9
5.4 执行标定	10
5.5 保存结果	12
6. 应用标定结果	13
7. 常见问题	18
8. 总结	18

1. 前言

相机标定是自瞄 solvePnP 解算中的关键，主要用于获取相机的内参矩阵、畸变系数，从而在图像中建立像素坐标与真实世界坐标的映射关系。

MATLAB 提供了 Camera Calibrator Toolbox，可以方便地完成标定实验。

2. 标定原理简述

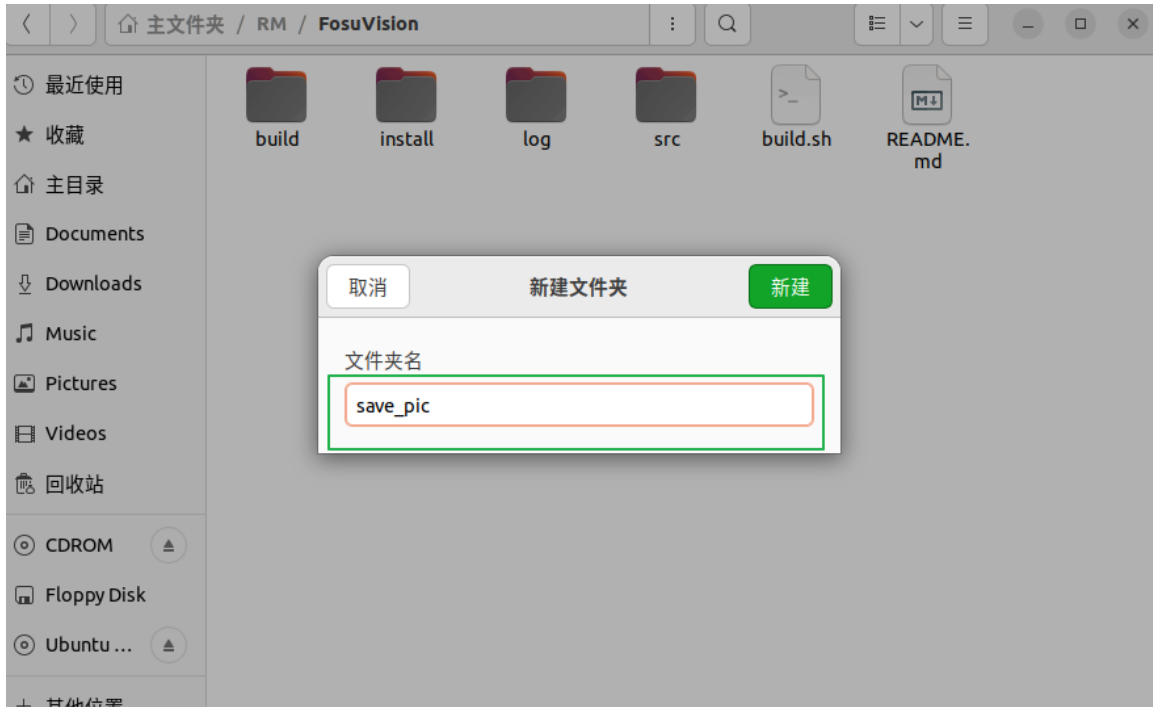
- 内参：描述相机成像特性：焦距 f_x 、 f_y ，主点 (c_x, c_y) 。
- 外参：表示相机坐标系到世界坐标系的旋转和平移。
- 畸变：镜头引起的径向、切向畸变，需要通过标定获得参数进行矫正。

3. 实验准备

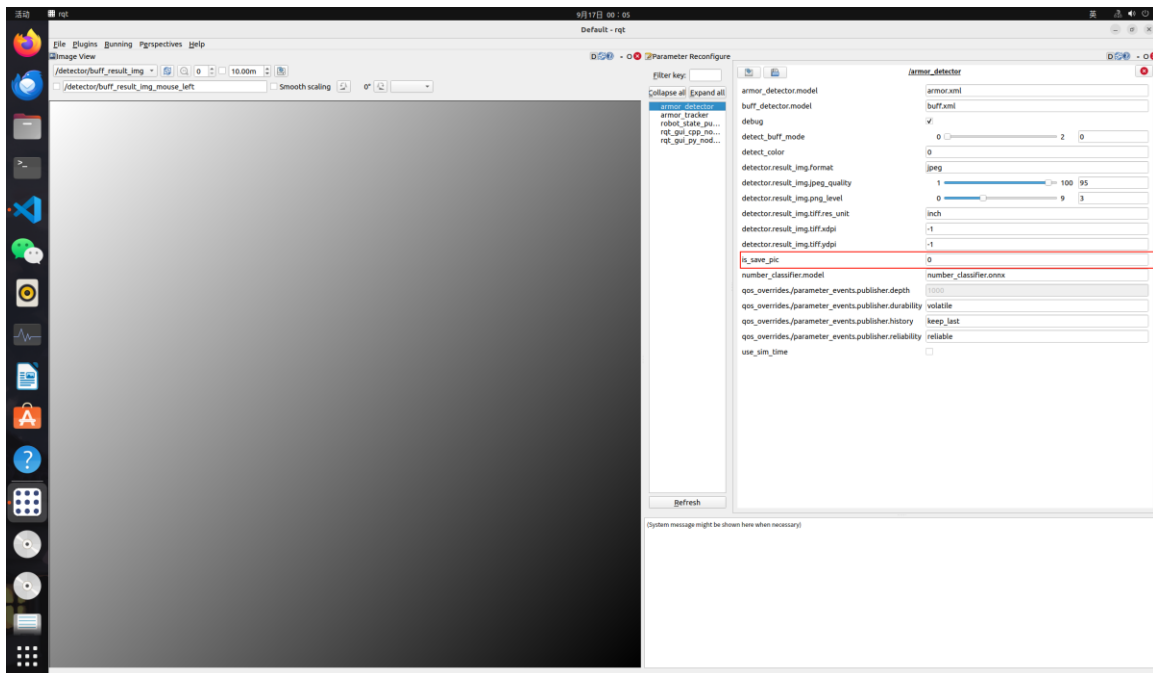
- 软件：MATLAB
- 硬件：需要标定的相机（迈德卫视、海康工业相机等）
- 标定工具：棋盘格标定板（我们使用的是 15mm 的）

4. 图像采集

1. 找一辆有相机的车车，打开小电脑，找到自瞄项目，在 src 目录下新建 save_pic 文件夹用于保存拍摄到的照片。



2. 运行代码，打开 rqt 并将 is_save_pic 置为 1



3. 采集 100-150 张棋盘格图片，确保棋盘在画面中的位置和角度多样：

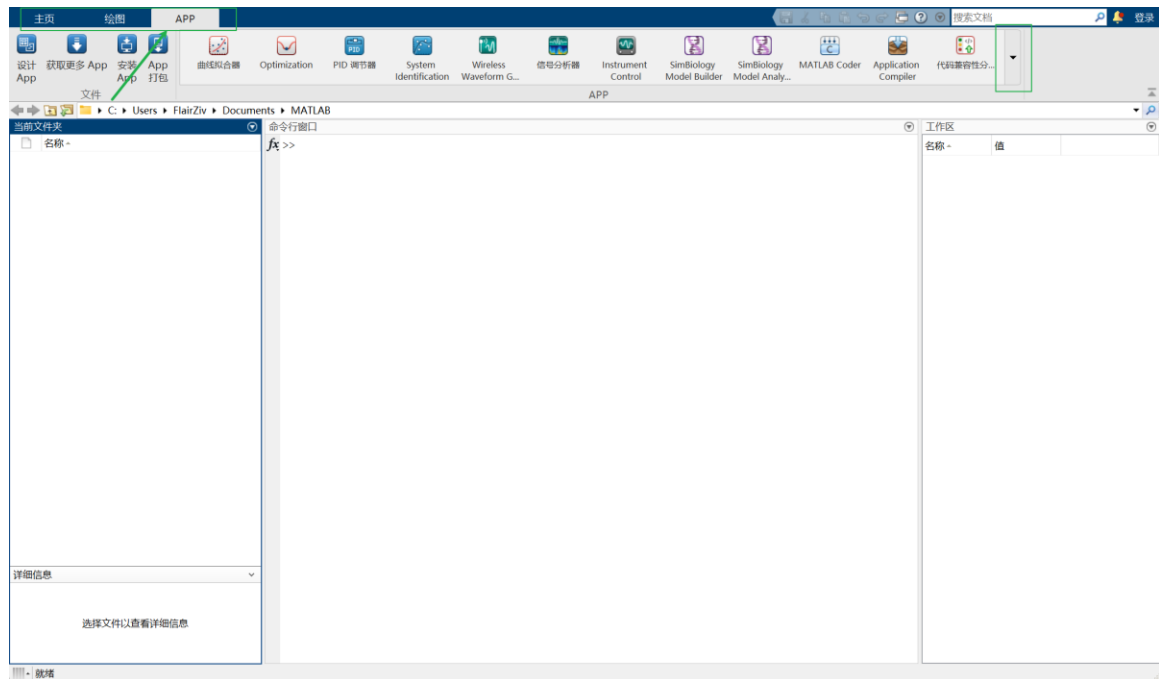
- 棋盘放在图像中心、四角、旋转一定角度
- 棋盘部分倾斜，以保证覆盖更多成像情况

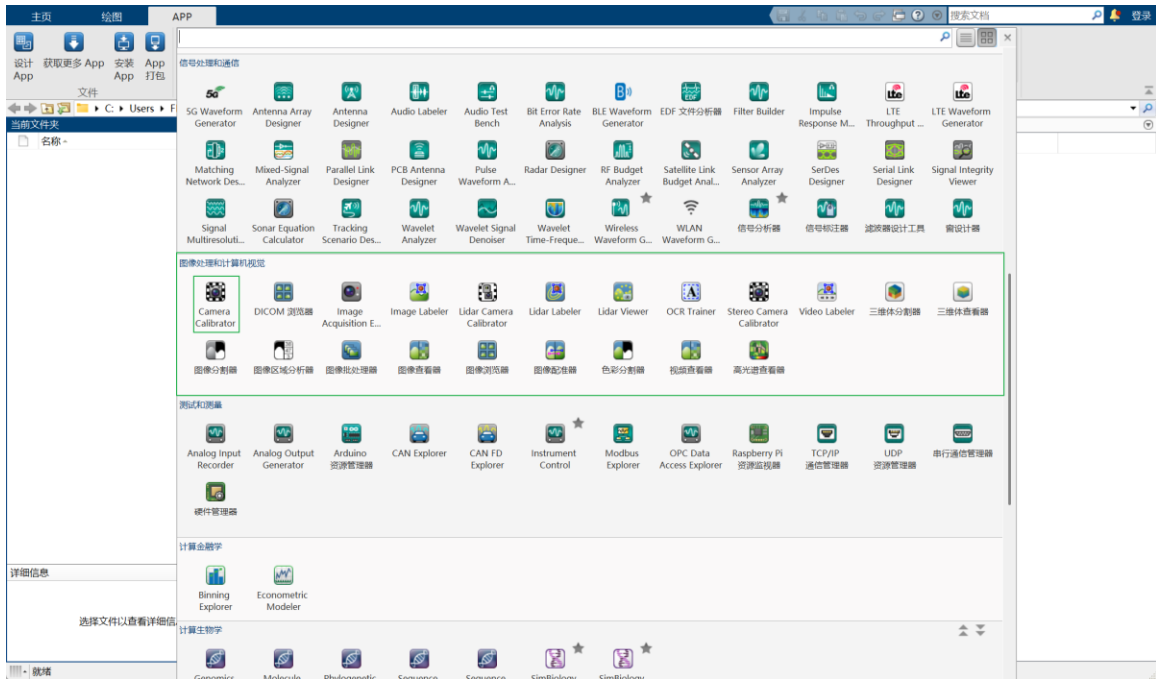
4. 采集的图片保存到一个文件夹，如：/home/rm/FosuVision/save_pic/img_

5. 使用 MATLAB 标定工具箱

5.1 打开标定工具

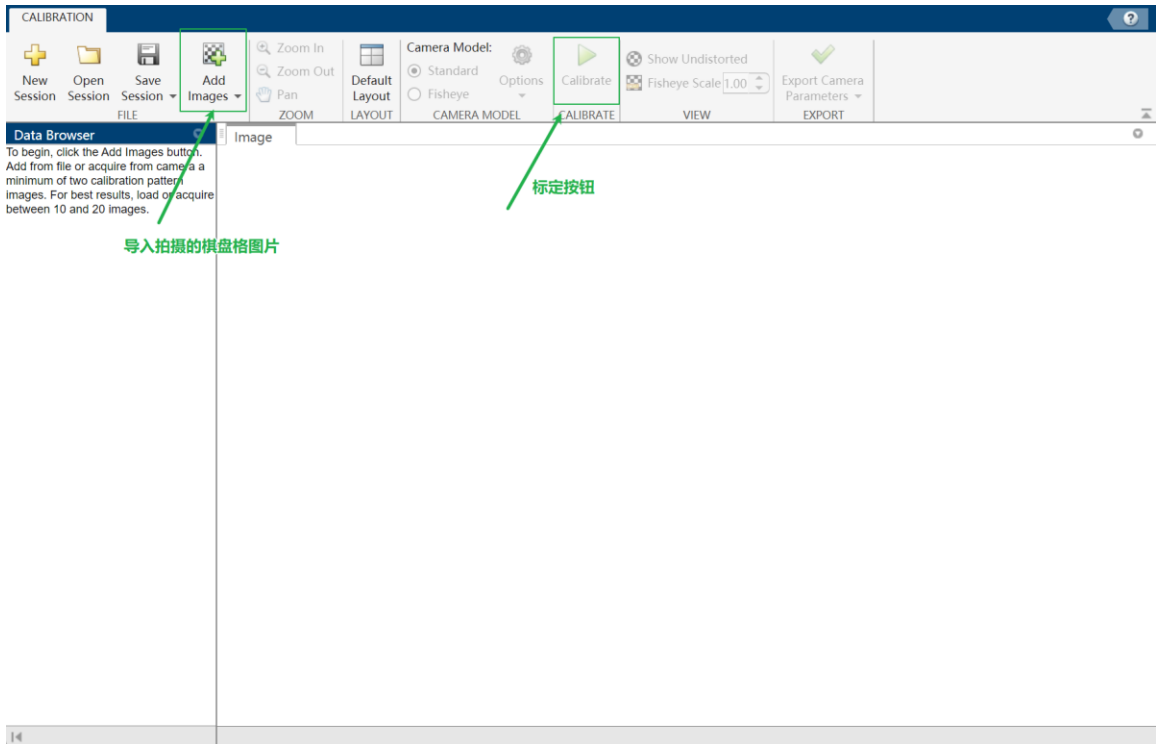
在 MATLAB 点击 APP，打开下拉导航按钮，找到 Camera Calibrator



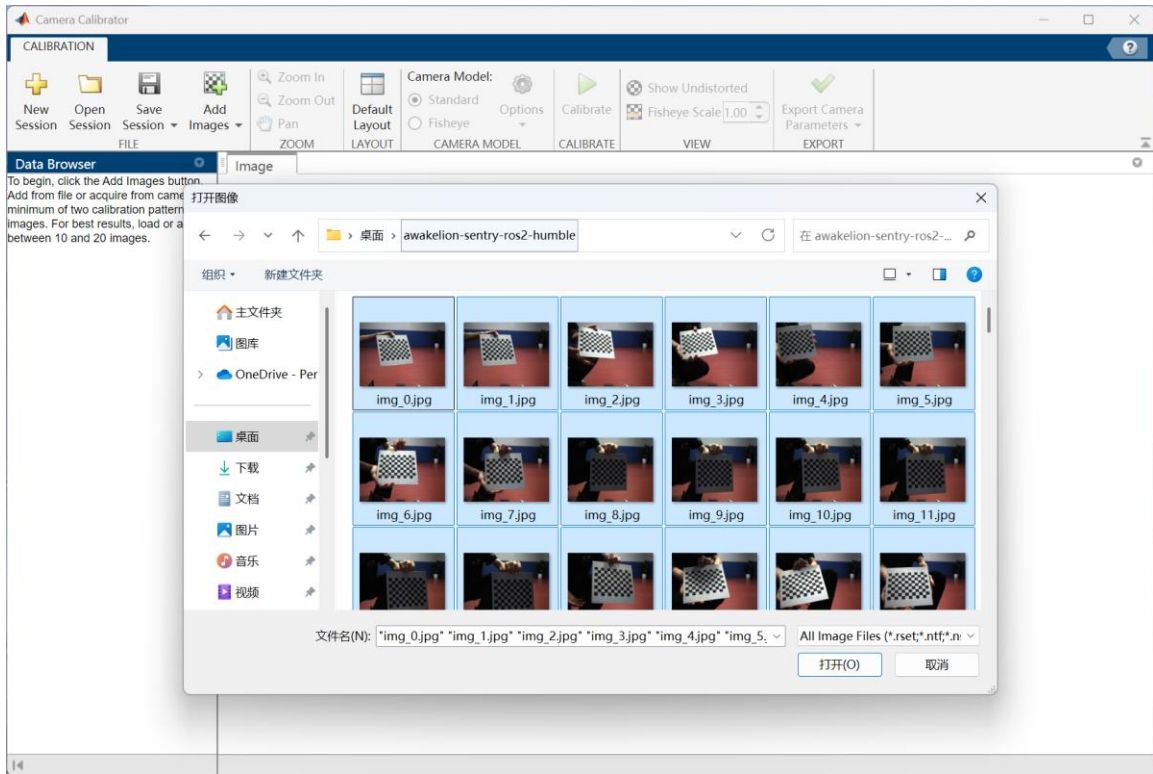


5.2 导入图像

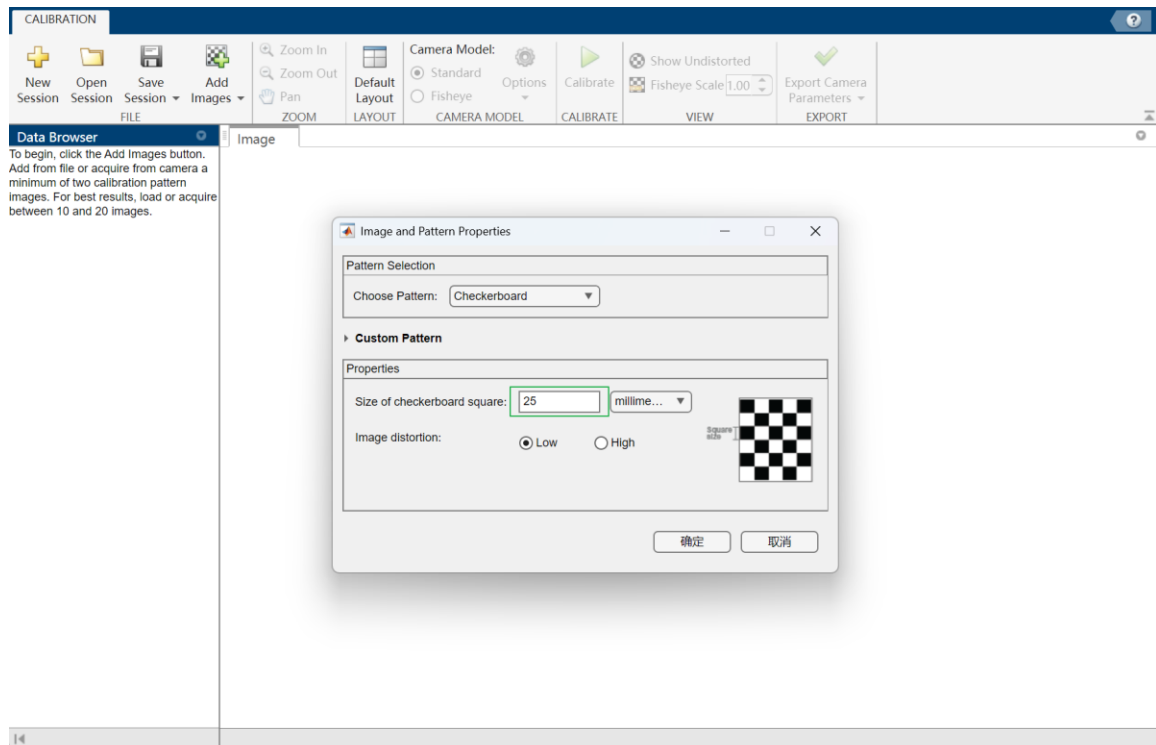
1. 在工具箱界面选择 Add Images



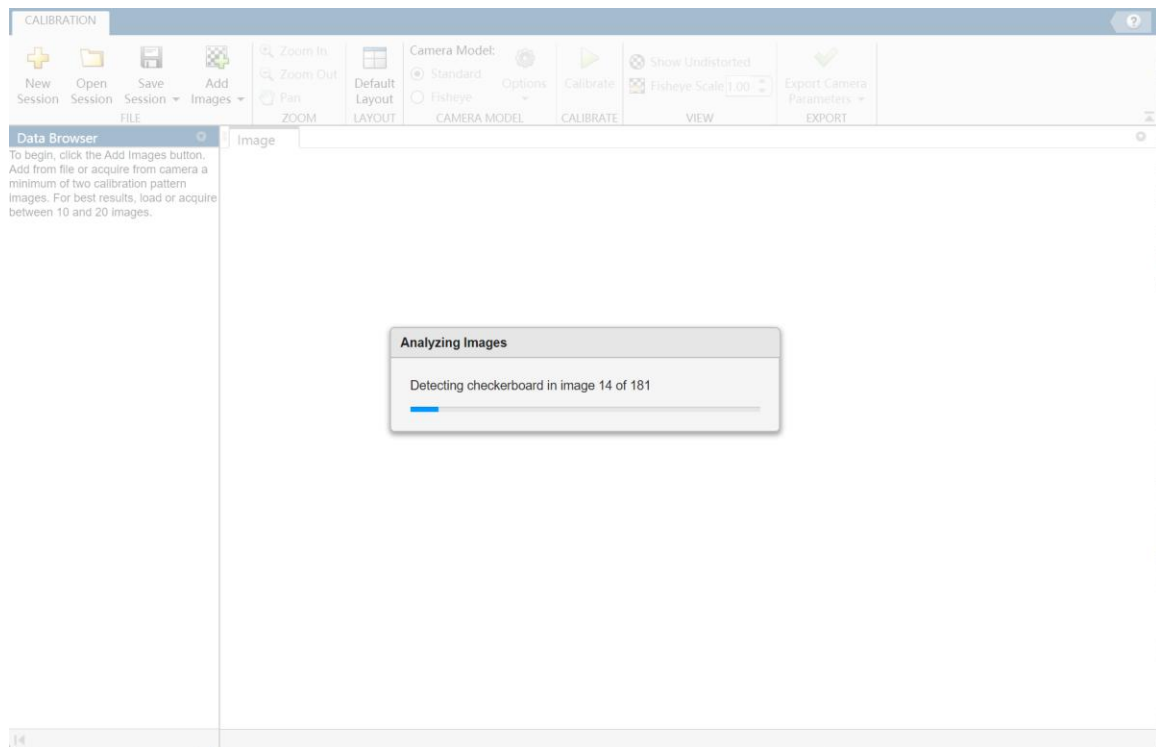
2. 选择采集的棋盘格图像文件夹，点击一张照片，ctrl + a 全选，点击打开即可导入



3. 输入棋盘格参数（例如：每个方格大小 15 mm）



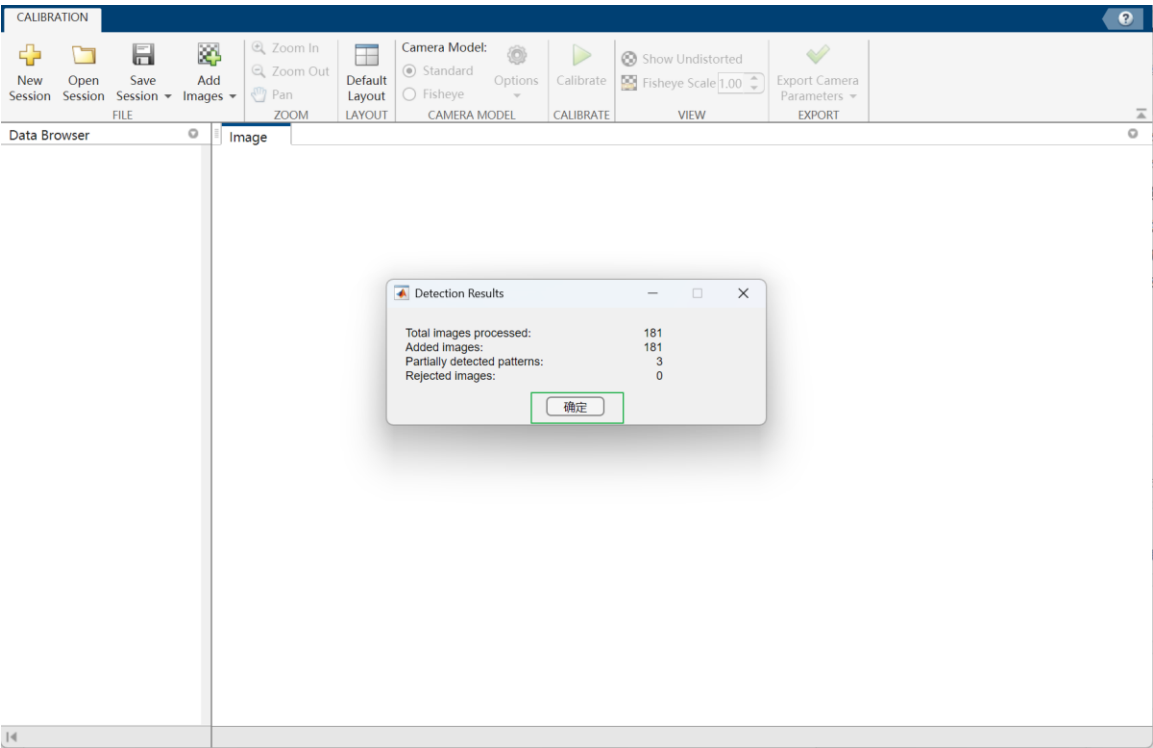
等待进度条



5.3 自动检测角点

MATLAB 会自动检测每张图像中的角点（棋盘格交点），结果会在界面显示。

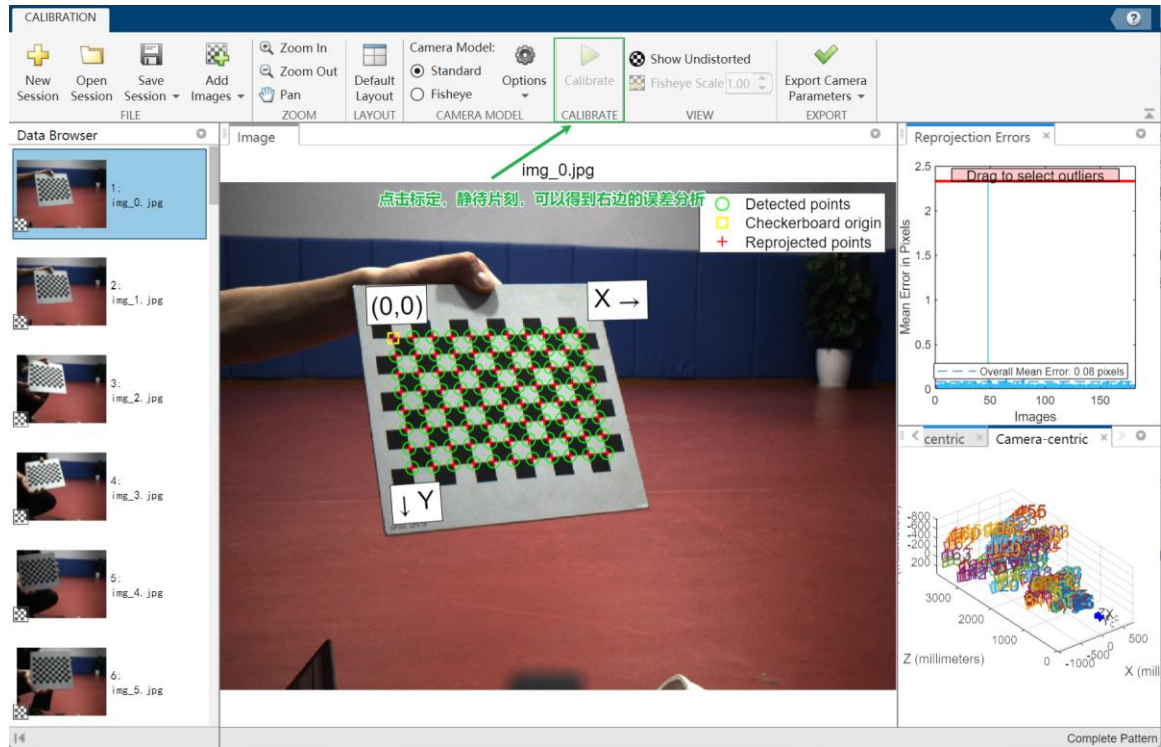
如果有未识别的图像，可以手动删除。

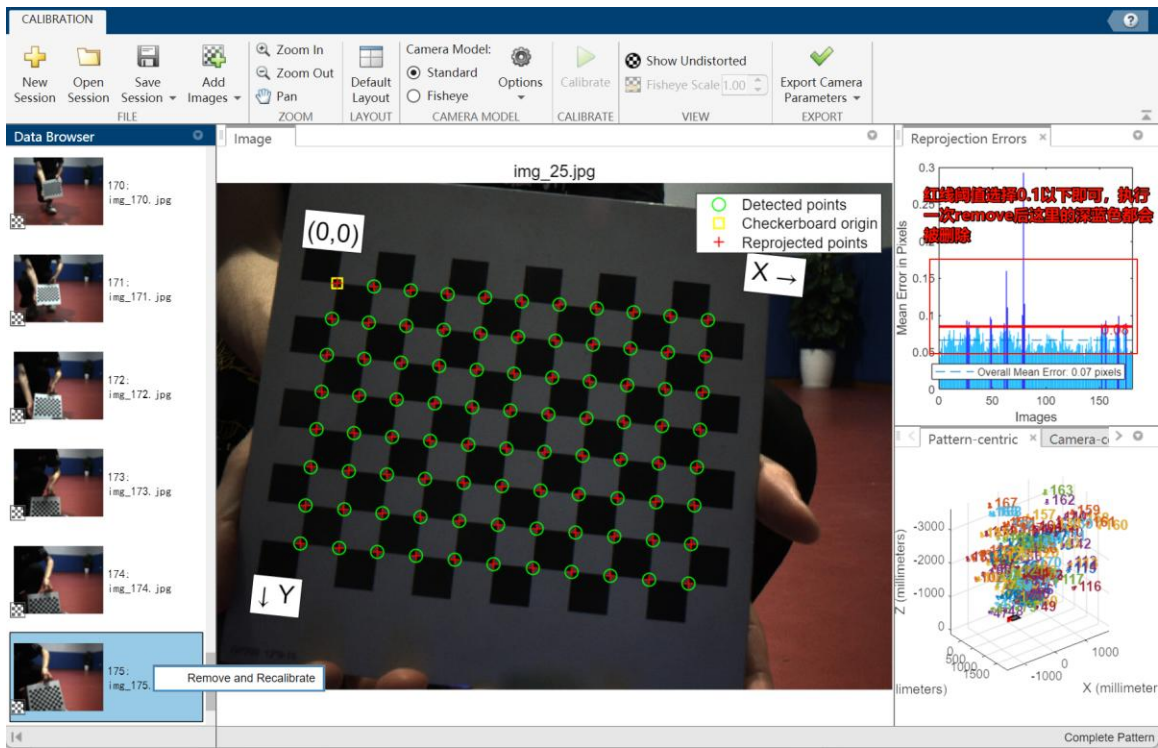
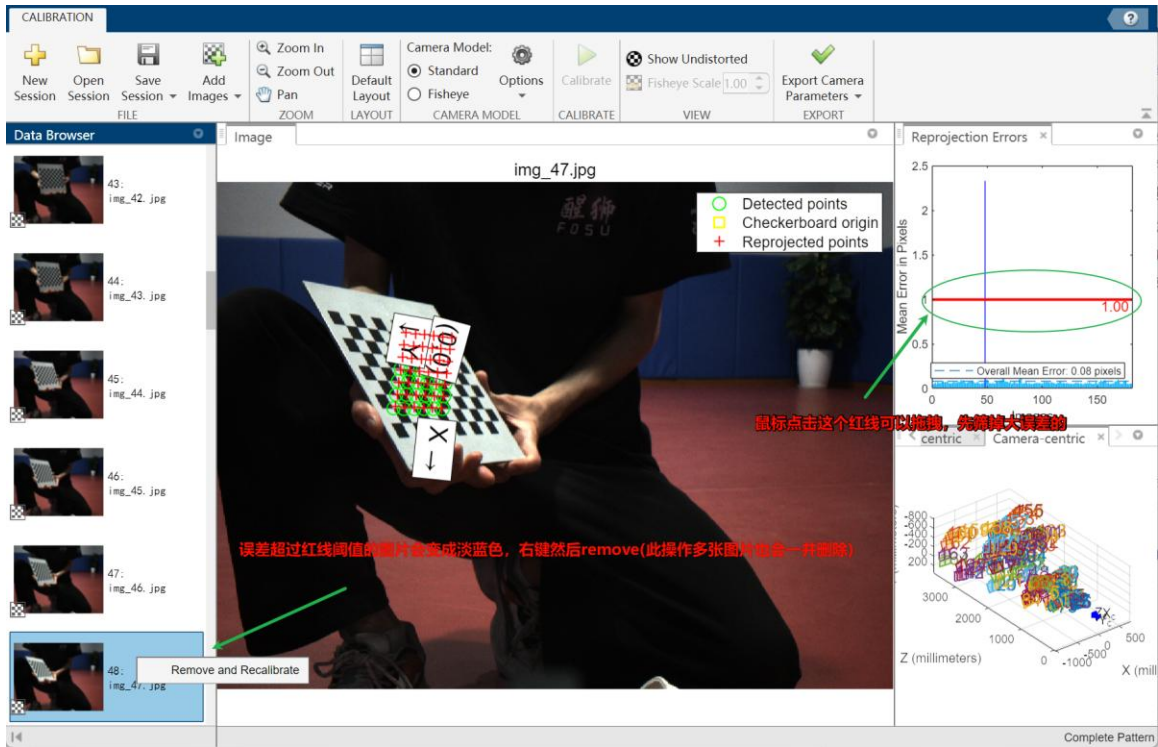


5.4 执行标定

点击 Calibrate 按钮，MATLAB 会计算相机的内参、畸变系数，并给出重投影误差。

- 误差越小 (<0.1 像素)，标定结果越好
- 如果误差过大，需增加更多不同角度的图像

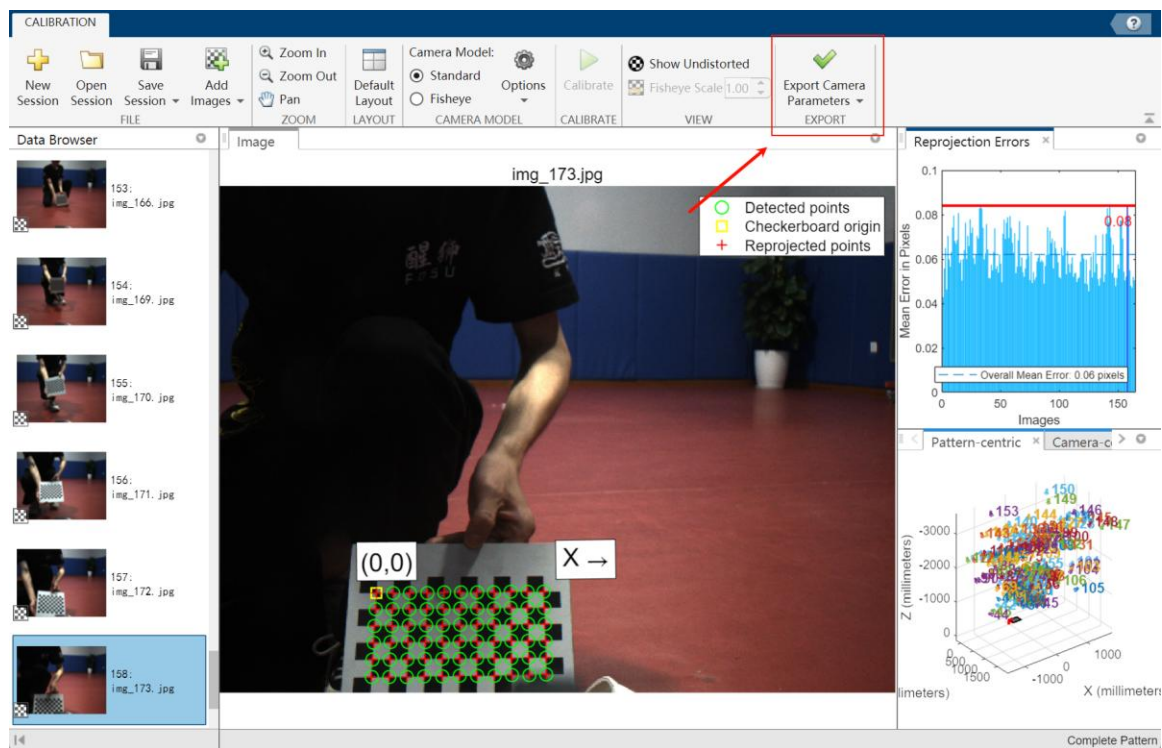


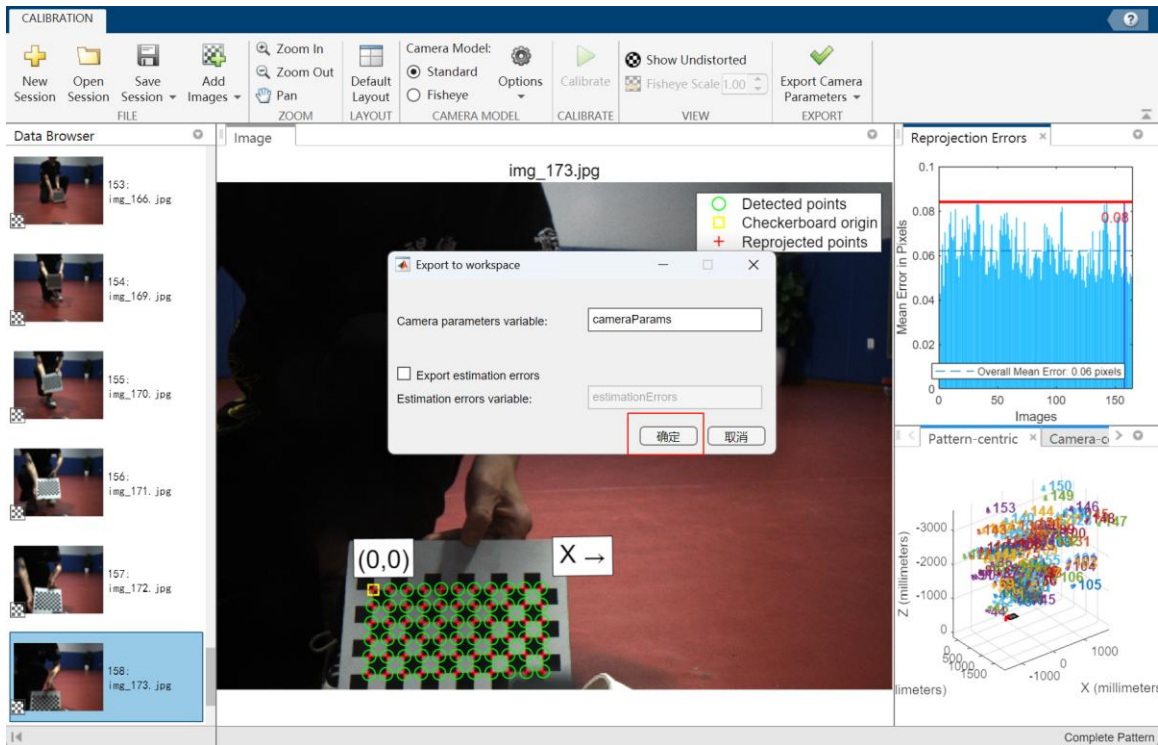


5.5 保存结果

在工具箱中点击 Export Camera Parameters，保存为一个 MATLAB 对象：

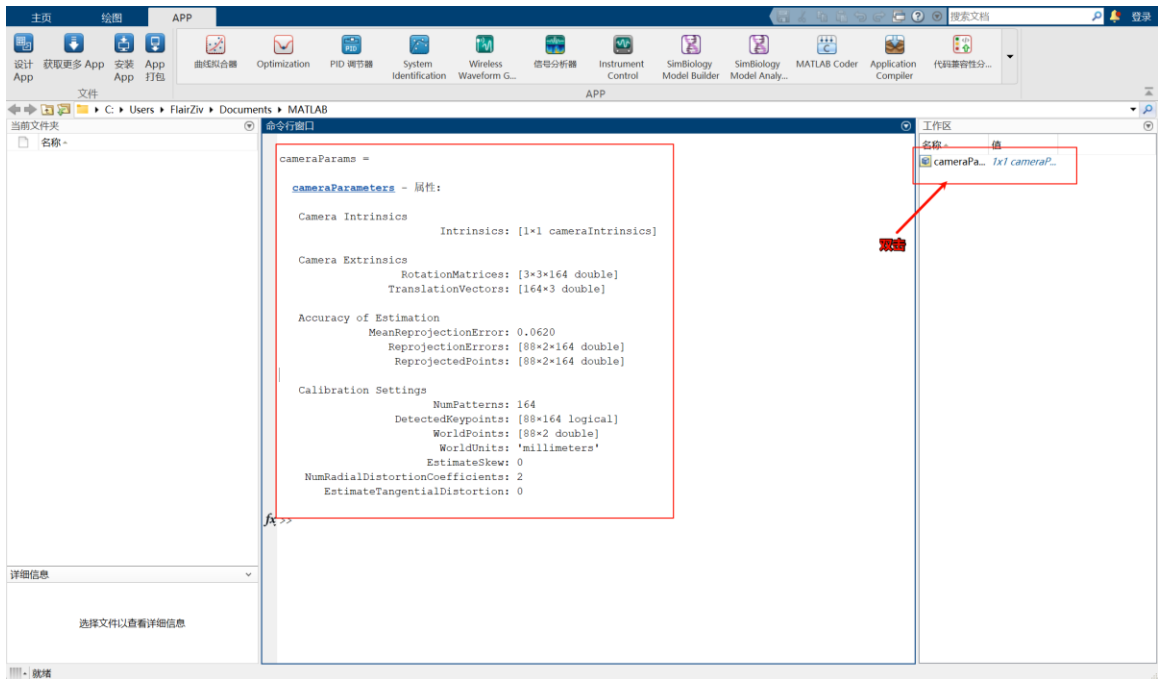
cameraParams



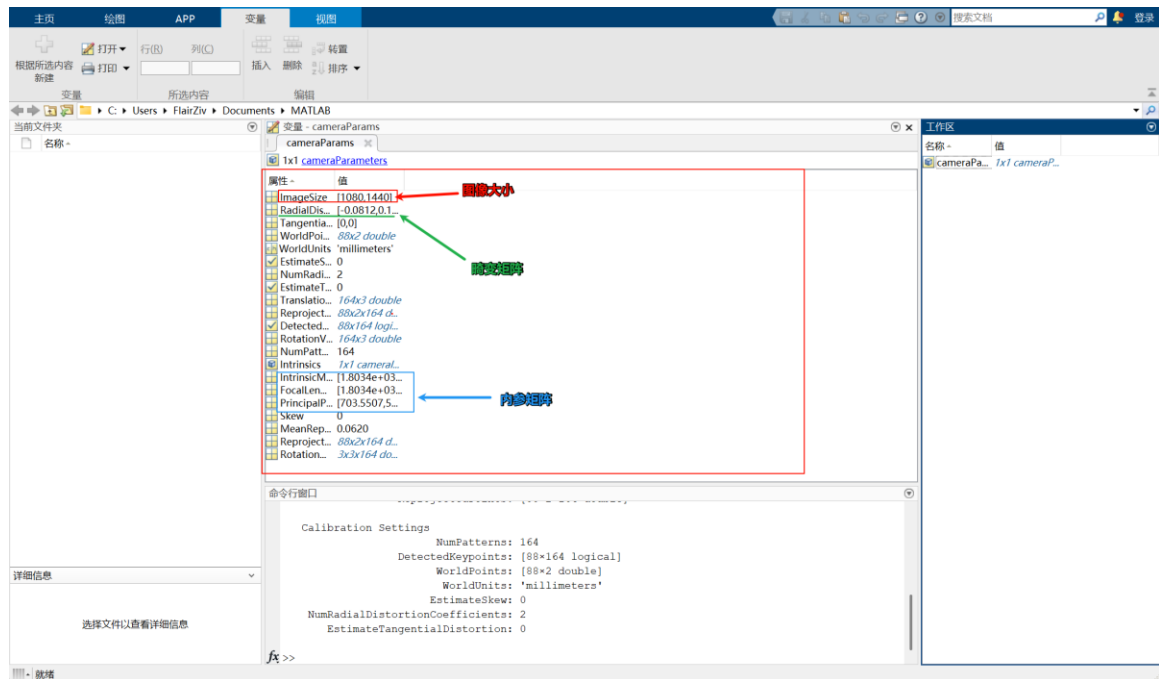


6. 应用标定结果

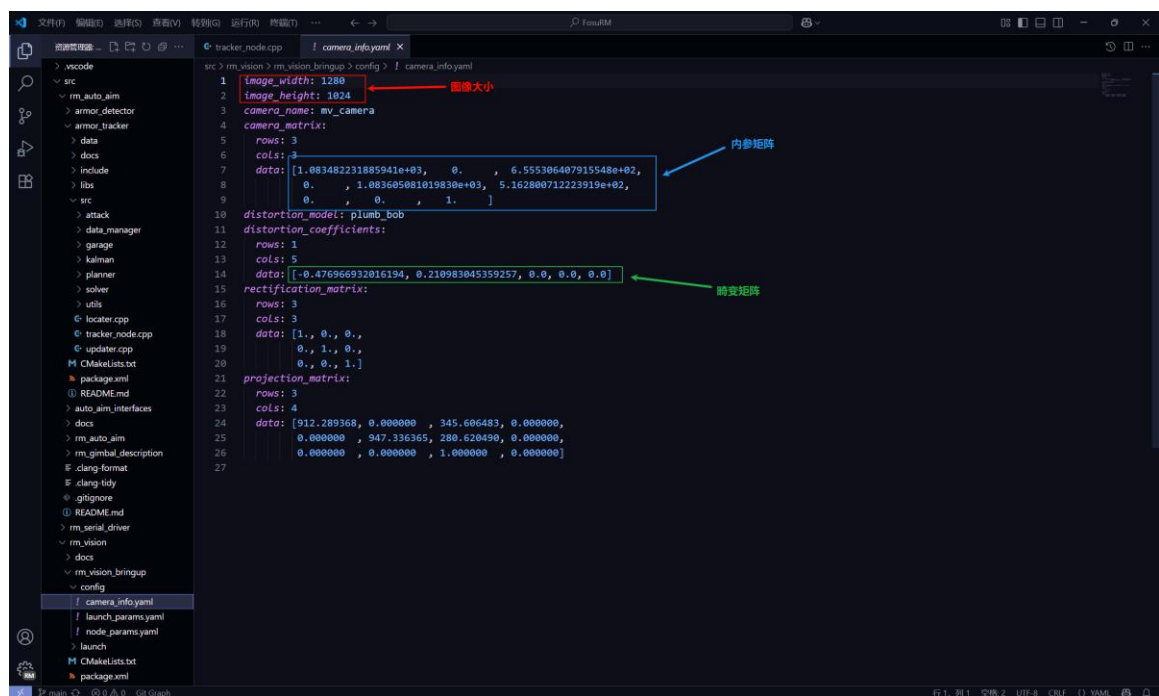
标定完成后，切换回 最开始打卡 matlab 的界面



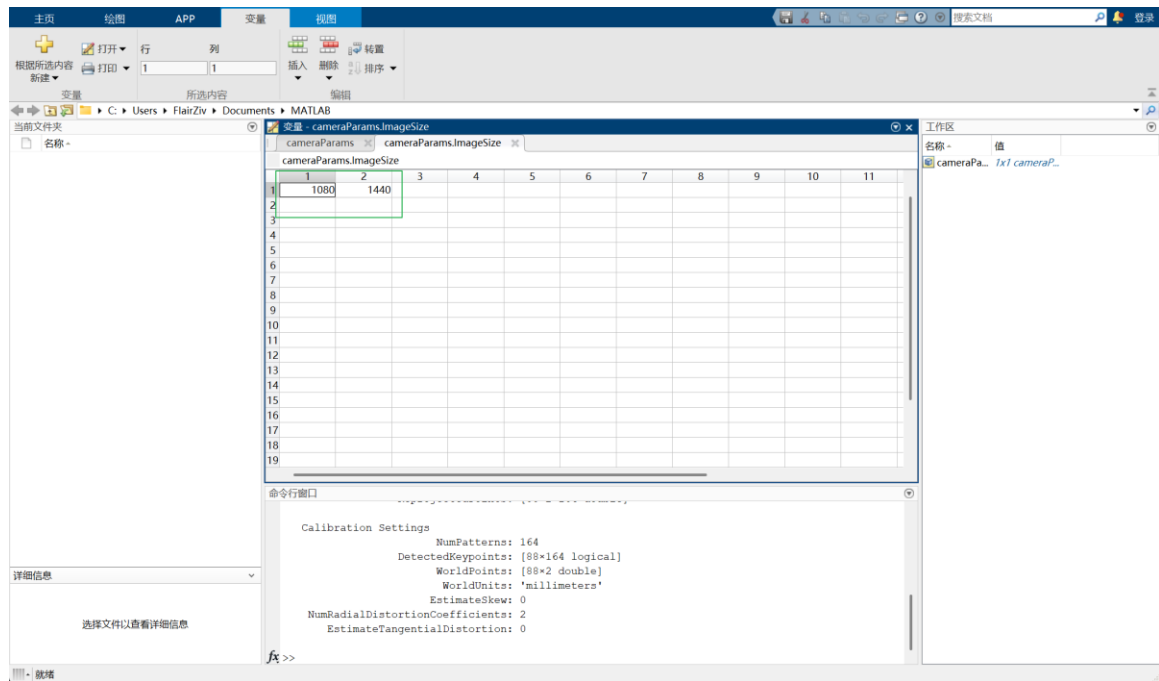
双击名字可以详细查看数据，然后把数据赋值进代码



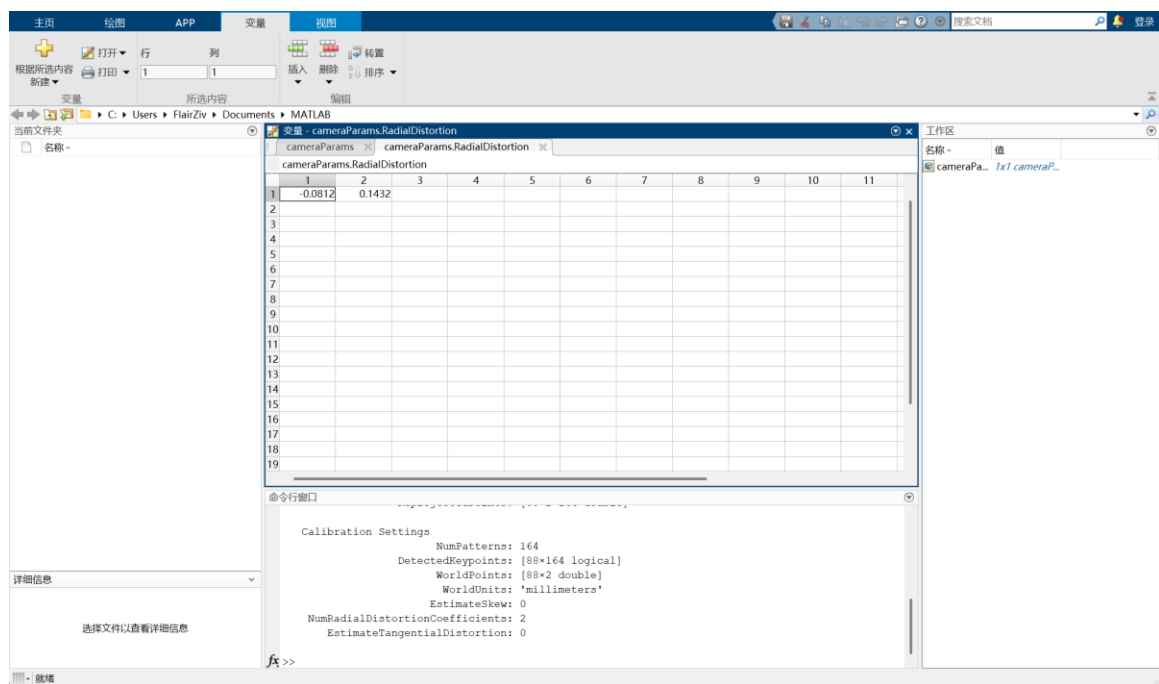
/home/rm/FosuVision /src/rm_vision/rm_vision_bringup/config/camera_info.yaml



图片大小调换顺序填入



畸变矩阵按顺序填入



主窗 视图 APP 变量 视图

根据所选内容 新建 打开 行 列 插入 删除 排序 转置

变量 所选内容 编辑

当前文件夹 名称

变量: cameraParams

cameraParams cameraParams.IntrinsicMatrix

1x1 cameraParameters

属性 值

- ImageSize [1080,1440]
- RadialDis... [-0.0812,0.1...
- Tangentia... [0,0]
- WorldPol... 88x2 double
- WorldUnits 'millimeters'
- EstimateS... 0
- NumRadi... 2
- EstimateT... 0
- Translati... 164x3 double
- Reproject... 88x2x164 d...
- Detected... 88x164 logi...
- RotationV... 164x3 double
- NumPatt... 164
- Intrinsics 1x1 camera...
- LocalLen... [1.8034e+03...
- PrincipalP... [703.5507,5...
- Skew 0
- MeanRep... 0.0620
- Reproject... 88x2x164 d...
- Rotation... 3x3x164 do...

命令窗口

Calibration Settings

NumPatterns: 164

DetectedKeypoints: [88x164 logical]

WorldPoints: [88x2 double]

WorldUnits: 'millimeters'

EstimateSkew: 0

NumRadialDistortionCoefficients: 2

EstimateTangentialDistortion: 0

fx >>

工作区

名称 值

cameraPa... 1x1 cameraP...

详细信息

选择文件以查看详细信息

就绪

这里显示了完整帧内参矩阵需要按3*3的矩阵转置输入代码

这里将内参矩阵内参矩阵转置，按照代码即可

主窗 视图 APP 变量 视图

根据所选内容 新建 打开 行 列 插入 删除 排序 转置

变量 所选内容 编辑

当前文件夹 名称

变量: cameraParams

cameraParams cameraParams.IntrinsicMatrix

cameraParams cameraParams.IntrinsicMatrix

1 1.8034e+... 0 0

2 0 1.8043e+... 0

3 703.5507 565.7767 1

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

命令窗口

Calibration Settings

NumPatterns: 164

DetectedKeypoints: [88x164 logical]

WorldPoints: [88x2 double]

WorldUnits: 'millimeters'

EstimateSkew: 0

NumRadialDistortionCoefficients: 2

EstimateTangentialDistortion: 0

fx >>

工作区

名称 值

cameraPa... 1x1 cameraP...

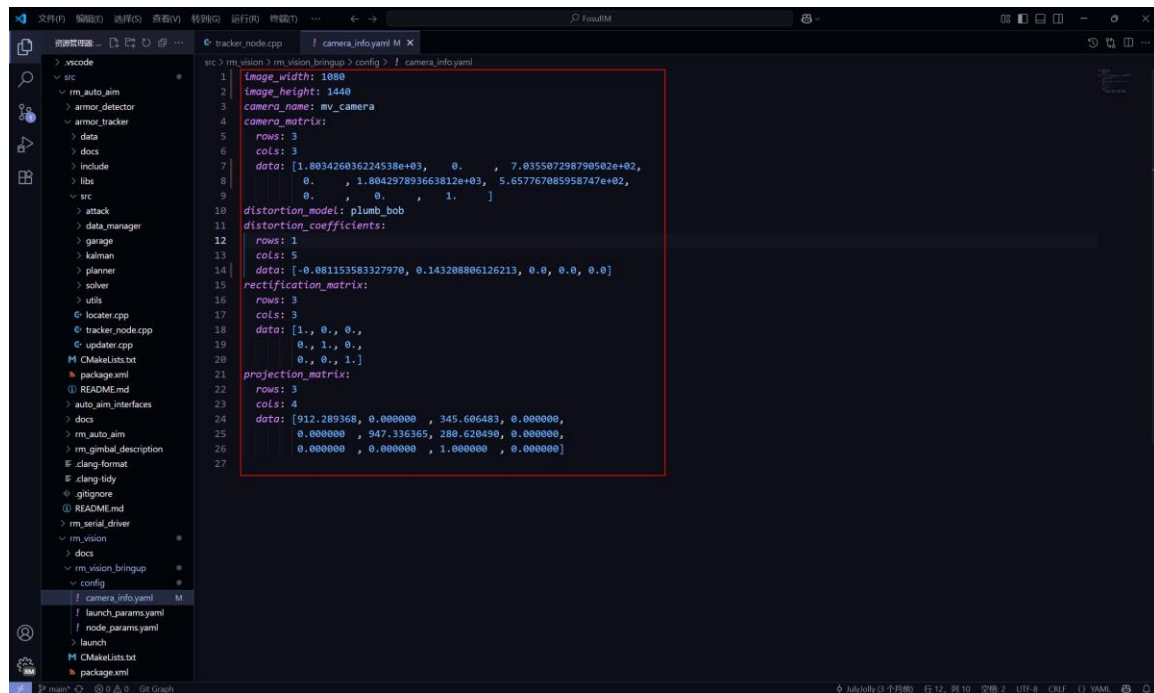
详细信息

选择文件以查看详细信息

就绪

转置成两行三列

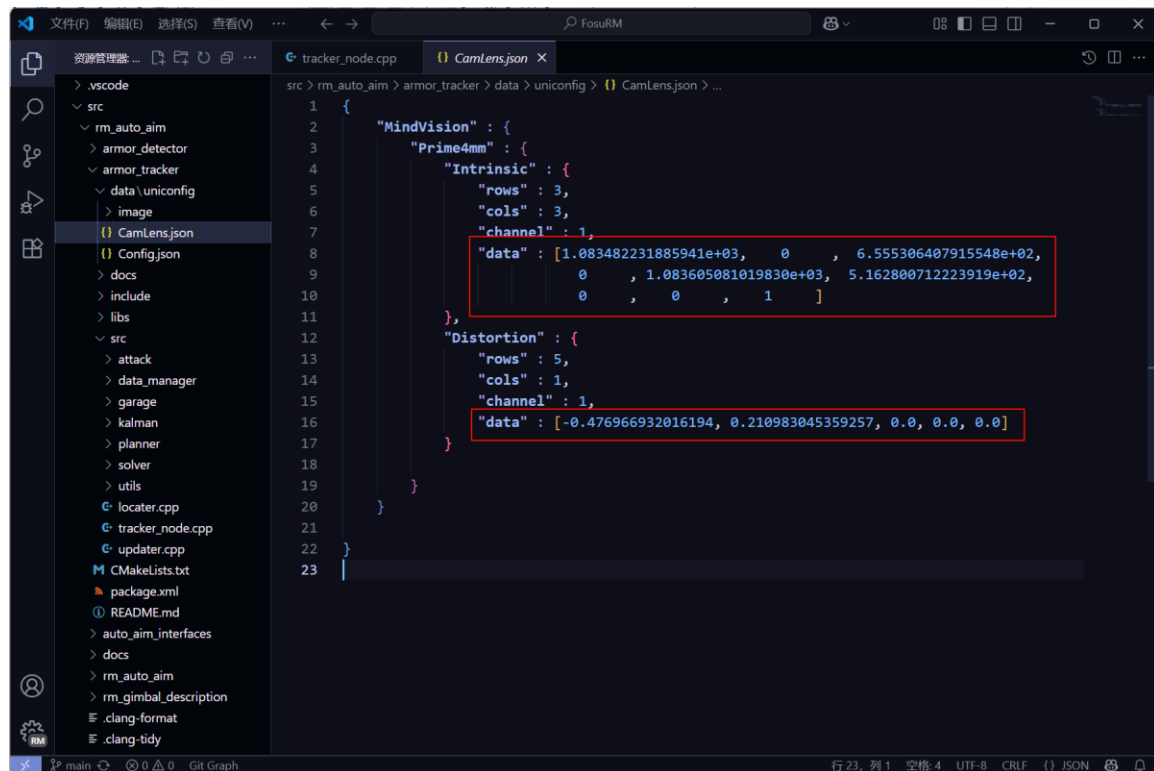
这里是修改后的



```
1 image_width: 1080
2 image_height: 1440
3 camera_name: mv_camera
4 camera_matrix:
5   rows: 3
6   cols: 3
7   data: [1.893426036224538e+03, 0., 7.035507298790502e+02,
8         0., 1.804297893663812e+03, 5.657767085958747e+02,
9         0., 0., 1.]
10 distortion_model: plumb_bob
11 distortion_coefficients:
12   rows: 1
13   cols: 5
14   data: [-0.081153583327970, 0.143288806126213, 0.0, 0.0, 0.0]
15 rectification_matrix:
16   rows: 3
17   cols: 3
18   data: [1., 0., 0.,
19         0., 1., 0.,
20         0., 0., 1.]
21 projection_matrix:
22   rows: 3
23   cols: 4
24   data: [912.289368, 0.000000, 345.606483, 0.000000,
25         0.000000, 947.336365, 280.620490, 0.000000,
26         0.000000, 0.000000, 1.000000, 0.000000]
```

最后记得修改 Json 的参数路径在:

/home/rm/FosuVision/src/rm_auto_aim/armor_tracker/data\uniconfig/CamLens.json



```
1 {
2   "MindVision" : {
3     "Prime4mm" : {
4       "Intrinsic" : {
5         "rows" : 3,
6         "cols" : 3,
7         "channel" : 1,
8         "data" : [1.083482231885941e+03, 0, 6.555306407915548e+02,
9                 0, 1.083605081019830e+03, 5.162800712223919e+02,
10                0, 0, 1]
11       },
12       "Distortion" : {
13         "rows" : 5,
14         "cols" : 1,
15         "channel" : 1,
16         "data" : [-0.476966932016194, 0.210983045359257, 0.0, 0.0, 0.0]
17       }
18     }
19   }
20 }
21
22
23
```

7. 常见问题

- 图像颠倒：确保标定时相机正放
- 角点无法检测到：确保棋盘格图像清晰且完整
- 误差过大：增加采集图片数量，覆盖更多角度
- 相机镜头更换后：必须重新标定
- 多久标定一次：一个月

8. 总结

通过本次实验，大家应能够完成：

1. 运行代码保存采集到的照片
2. 使用 MATLAB 采集棋盘格图像
3. 使用 Camera Calibrator Toolbox 完成标定
4. 获取相机内参与畸变参数