手眼标定

1.运行get_data_new.py，获取标定需要的图片和位姿数据

注意从UR5机械臂的示教器中读取一个，处在正常位置时的xyz和rx,ry,rz，然后以这个数据为参考，将下面的测试范围值设置为一个相对合适的范围。如果设置不合理，会使机械臂移动到一些很奇怪的姿势。

```python
tcp_host_ip = '192.168.3.10'  # IP和端口，机器人控制器额的IP地址
workspace_limits = np.asarray([[-0.05, 0], [-0.35, -0.30], [0.35, 0.40]])
      #定义了机器人工作空间的范围，是一个3x2的NumPy数组，表示x、y、z三个方向上的最小值和最大值。
calib_grid_step = 0.02
      #标定网格的步长
tool_orientation = [2 * np.pi / 3, 2 * np.pi / 3, 0]
```

添加的旋转扰动要稍微大一些，大概在接近30度的值，否则求不出手眼矩阵，报错如下：

[ERROR:0@1.733] global calibration_handeye.cpp:335 calibrateHandEyeTsai Hand-eye calibration failed! Not enough informative motions--include larger rotations.

```python
# 添加小的随机旋转扰动（单位：弧度）
    random_rotation = np.random.uniform(-0.5, 0.5, 3)
```

2.运行xyz_npy2excel.py，将每一个保存的位姿数据以.npy结尾的形式转化为.xlsx

3.运行test.py，计算手眼矩阵

```
R_tool 的大小是：(27, 3, 3)
t_tool 的大小是：(27, 3, 1)
rotation_matrices 的大小是：(9, 3, 3)
translation_vectors 的大小是：(9, 3, 1)
```

有些拍摄的标定板图片不完整，因此提取不出完整角点，也就不会得到目标相对于相机的位姿，因此我们要在.xlsx文件里，删除那些没有提取出完整角点的图片对应的位姿。

我们从这里可以看到那些图片是不可用的，比如12.png，然后就去.xlsx里面删除第13行（图片命名以0开始，.xlsx以1开始）

```
0/26
False
chessboard not found: ./IMG/12.png
1/26
False
chessboard not found: ./IMG/14.png
2/26
False
chessboard not found: ./IMG/16.png
3/26
False
chessboard not found: ./IMG/8.png
4/26
True
5/26
False
chessboard not found: ./IMG/6.png
6/26
False
chessboard not found: ./IMG/23.png
7/26
False
chessboard not found: ./IMG/13.png
8/26
False
chessboard not found: ./IMG/11.png
9/26
True
10/26
False
chessboard not found: ./IMG/26.png
11/26
False
chessboard not found: ./IMG/7.png
12/26
False
chessboard not found: ./IMG/15.png
13/26
False
chessboard not found: ./IMG/17.png
14/26
True
15/26
True
16/26
False
chessboard not found: ./IMG/9.png
17/26
True
18/26
True
19/26
False
chessboard not found: ./IMG/22.png
20/26
True
21/26
False
chessboard not found: ./IMG/3.png
```

最后要保证相应矩阵的维度保持一致。

```
R_tool 的大小是: (9, 3, 3)
t_tool 的大小是: (9, 3, 1)
rotation_matrices 的大小是: (9, 3, 3)
translation_vectors 的大小是: (9, 3, 1)
```

最后使用cv2.calibrateHandEye(R_tool, t_tool, rotation_matrices, translation_vectors, cv2.CALIB_HAND_EYE_PARK)计算出手眼矩阵的旋转和平移

标定逻辑

使用的是cv2.calibrateHandEye计算手眼矩阵

R, t = cv2.calibrateHandEye(R_gripper2base, t_gripper2base, R_target2cam, t_target2cam, method)

- `R_gripper2base`：这是一个列表，列表中的每个元素是一个 3x3 的旋转矩阵，表示机器人末端执行器（夹爪）在不同位姿下相对于机器人基座坐标系的旋转。
- `t_gripper2base`：同样是一个列表，每个元素为 3x1 的平移向量，代表机器人末端执行器在不同位姿下相对于机器人基座坐标系的平移。
- `R_target2cam`：列表形式，每个元素是 3x3 的旋转矩阵，体现标定目标（如标定板）在不同位姿下相对于相机坐标系的旋转。
- `t_target2cam`：列表结构，每个元素为 3x1 的平移向量，意味着标定目标在不同位姿下相对于相机坐标系的平移。

1. 1.gripper2base通过获取机械臂末端位姿，并对其进行格式转换得到
2. 2.target2cam通过cv2.calibrateCamera先获得旋转向量，然后进行格式转换

可能问题

可能出现下面问题

问题描述：[ERROR:0@1.418] global calibration_handeye.cpp:335 calibrateHandEyeTsai Hand-eye calibration failed! Not enough informative motions--include larger rotations.

问题原因：采集的图片旋转量不足，特别是缺少足够大的旋转运动。

手眼标定需要机器人在空间中执行一系列运动，以获取相机相对于机器人末端的精确关系。若机器人的运动数据中缺乏足够的旋转变化，尤其是在各个轴向上的显著旋转，标定算法就无法准确计算出手眼关系。

解决方案：

1. **增加旋转运动：**

- 在数据采集过程中，让机器人执行更大的旋转运动。确保在三个轴（X、Y、Z）上都有足够的旋转角度变化。
- 例如，旋转角度最好能超过30度，这样能提供丰富的运动信息。

1. **多样化运动姿态：**

- 采集数据时，保证机器人末端执行多样化的姿态变化，包括平移和旋转。
- 避免只在小范围内运动或只在某一轴上运动。

1. **增加数据采集次数：**

- 采集更多的样本数据，一般来说，至少需要10组以上不同的姿态数据。
- 更多的数据能提高标定的稳定性和准确性。