**Lab3**

**1.Basic Part**

**1.1 Introduction**

**1.2 Experiment Setup**

**1.3 Result and Data Processing**

**1.3.1 Chessboard Image Acquisition with Varied View Angles**

(1) Select a chessboard image, open it on an iPad, and place it flat on the ground as a calibration reference.



1. {
2. "img\_height": 2560,
3. "img\_width": 1440,
4. "row": 12,
5. "column": 9,
6. "block\_size": "150 pixel",
7. "Warning": "the input of cv2.findChessboardCorners  is 11 and 8"
8. }

(2) Assemble the camera and lens, then develop a control script.

Utilizing the PySpin library to interface with FLIR camera functionalities, this script enables an automated workflow where pressing the spacebar triggers automatic image capture. The captured images are sequentially named and saved to a designated directory.

(3) Captured chessboard images from **multiple angles** under **three different camera heights**.

|  |  |
| --- | --- |
| Height 1 |  |
| Height 2 |  |
| Height 3 | 17 |

(4) Acquire images under the condition of a fixed angle but **different shooting distances**.

|  |  |
| --- | --- |
| Distance 1 | 8 |
| Distance 2 | 10 |
| Distance 3 | 11 |

These two categories of images were stored in separate folders respectively to facilitate subsequent data management and analysis.

**1.3.2 Single Camera Calibration with Chessboard and PLY Mesh Generation**

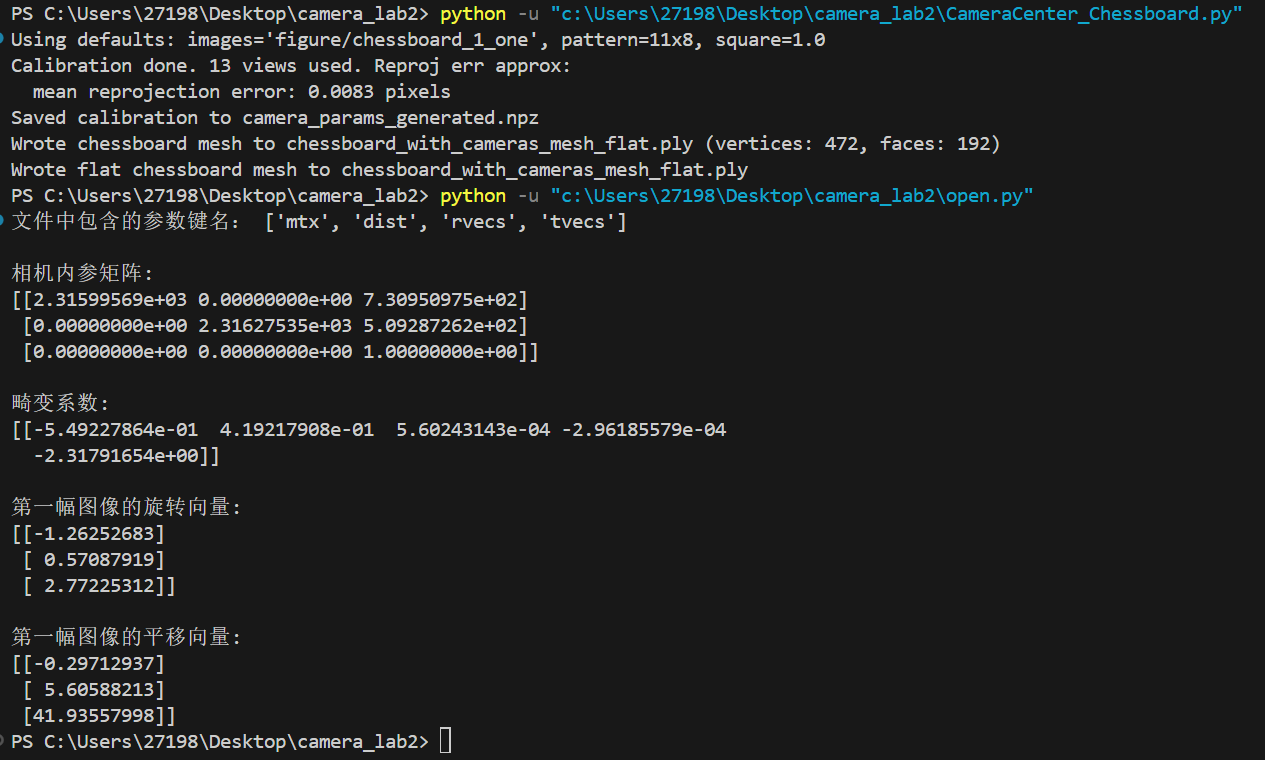
**(1) Chessboard Corner Detection and Data Collection**

Extract subpixel-accurate chessboard corners from images, and collects 3D world coordinates and 2D image coordinates for calibration.

**(2) Camera Intrinsic and Extrinsic Calibration**

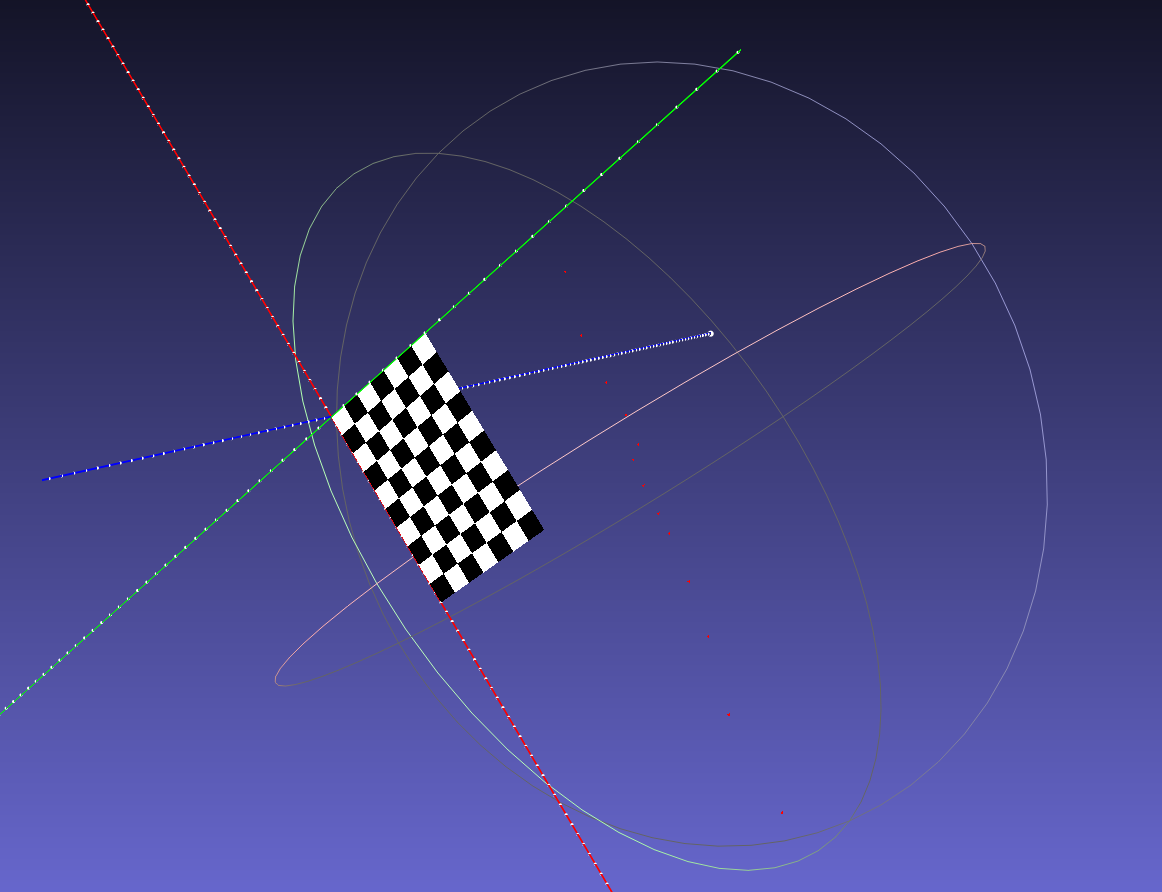
Use the 3D world coordinates of chessboard corners and 2D image pixel coordinates of chessboard corners collected in Phase 1 to compute the camera’s intrinsic and extrinsic parameters.

1. import numpy as np
2. calib\_params = np.load("camera\_params\_generated.npz")
3. print("文件中包含的参数键名：", calib\_params.files)
4. mtx = calib\_params["mtx"]          # 相机内参矩阵
5. dist = calib\_params["dist"]        # 畸变系数
6. rvecs = calib\_params["rvecs"]      # 旋转向量列表
7. tvecs = calib\_params["tvecs"]      # 平移向量列表
8. print("\n相机内参矩阵:")
9. print(mtx)
10. print("\n畸变系数:")
11. print(dist)
12. print("\n第一幅图像的旋转向量:")
13. print(rvecs[0])
14. print("\n第一幅图像的平移向量:")
15. print(tvecs[0])
16. calib\_params.close()



**(3) 3D PLY Mesh Generation**

Convert the calibrated chessboard and camera poses into a 3D PLY mesh file to visualize the chessboard and camera centers in MeshLab.

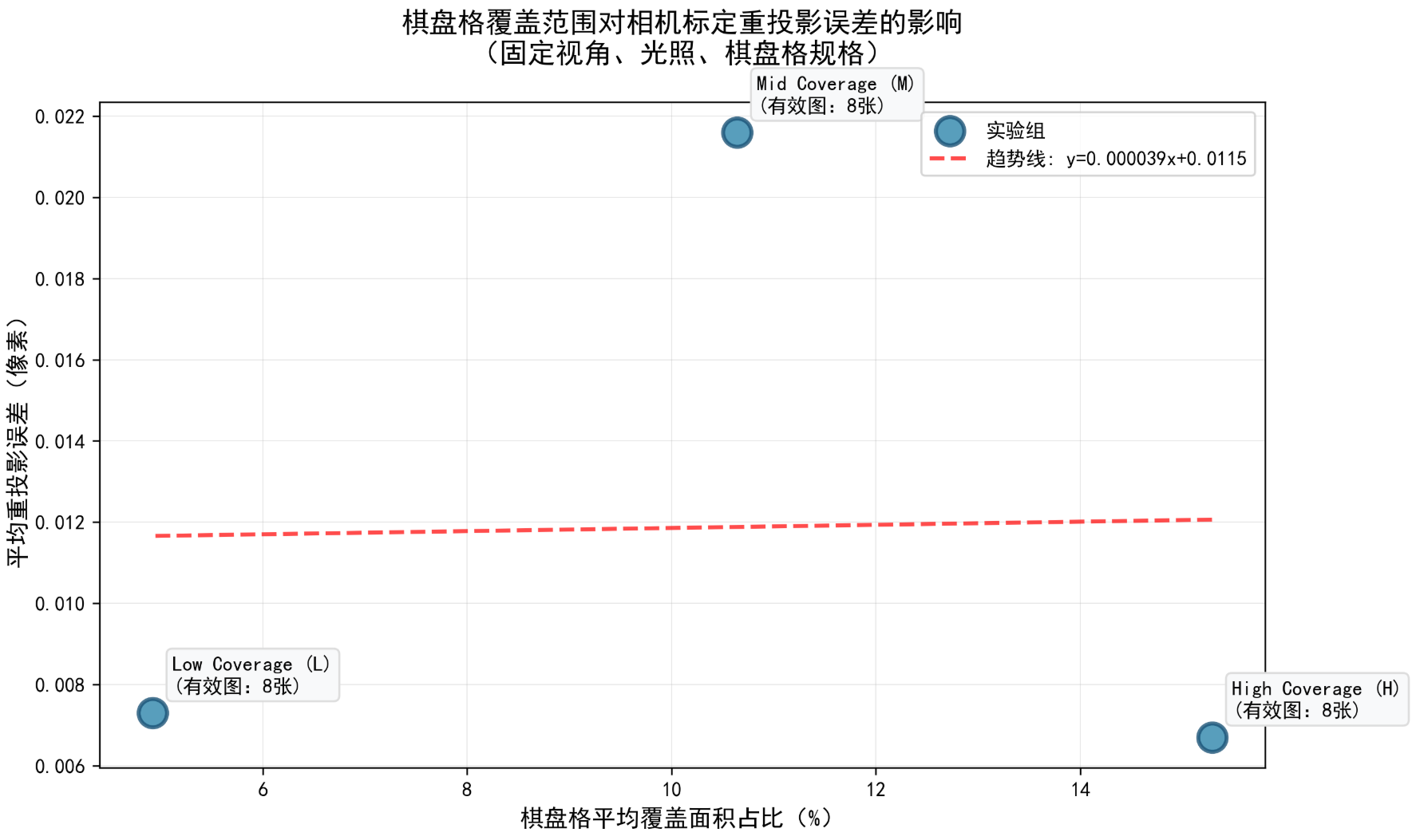


**1.3.3 Quantifying the Effect of Chessboard Coverage Ratio on Final Reprojection Error**

This controlled experiment aims to quantitatively examine how the chessboard coverage ratio (the percentage of the chessboard’s pixel area relative to the total image area) influences the final reprojection error in single-camera calibration, while strictly isolating this variable from other potential factors that could affect calibration accuracy. The core logic is to maintain all experimental conditions consistent except for the chessboard coverage ratio, allowing for a direct and reliable correlation between the variable and the outcome.

Three experimental groups are established to create distinct gradients of chessboard coverage ratio, with all other conditions strictly fixed.

|  |  |  |  |
| --- | --- | --- | --- |
| **Coverage** | **Low** | **Mid** | **High** |
| **Average Coverage Area Ratio** | **4.92%** | **10.64%** | **15.29%** |
| **Number of Valid Images** | **8** | **8** | **8** |
| **Average Reprojection Error** | **0.0073 pixels** | **0.0216 pixels** | **0.0067 pixels** |



**1.4 Analysis and Discussion**

The experimental result shows that the mid-coverage group has an abnormally high reprojection error. This is likely due to a combination of factors: uneven lighting on the chessboard, as seen in the images where illumination isn’t consistent, which messes up corner detection accuracy. Also, the camera-chessboard angle might not be optimal, causing perspective distortion that the calibration algorithm struggles to handle. Plus, mid-coverage images have the chessboard mostly in the central region, where lens distortion is less prominent, so the algorithm lacks enough edge data to properly model distortion, leading to inaccurate calibration and higher error.

**1.5 Conclusion**

In conclusion, for achieving accurate single-camera calibration, not only the chessboard coverage ratio but also factors such as uniform lighting, optimal camera-chessboard angles, and sufficient distribution of chessboard corners across the image (including edge regions) should be carefully considered.

**2.Bonus Part**

**3.Reference**