Name: Andrew Hayes
Student ID: 21321503

CT404

Name: Maxwell Maia
Student ID: 21236277

Lab Assignment 1: Camera Callibration

1 *P*-Matrix Estimation Using Provided Code

```
filename = rubiks.jpg
Click on the image to select 96 points and enter their 3D world coordinates.
Computed camera matrix P:
  -9.0391e-02  -1.9850e-02   4.9319e-02  -7.8258e-01
   8.8371e-03  -2.6519e-03   1.0434e-01  -6.0464e-01
  -1.8517e-05   2.5680e-05   2.7495e-05  -5.3967e-04
Intrinsic matrix K:
   2.0418e+03   1.7659e+02   1.4332e+03
            0   1.9977e+03   1.4999e+03
            0            0   1.0000e+00
Rotation matrix R:
  -0.783587  -0.619216   0.050626
   0.437028  -0.491447   0.753317
  -0.441586   0.612414   0.655706
Camera center (in world coordinates):
  -7.0008
   8.8862
   6.6135
>> |
```

Figure 1: Command window output showing he computed camera matrix P, the intrinsic matrix K, & the rotation matrix R

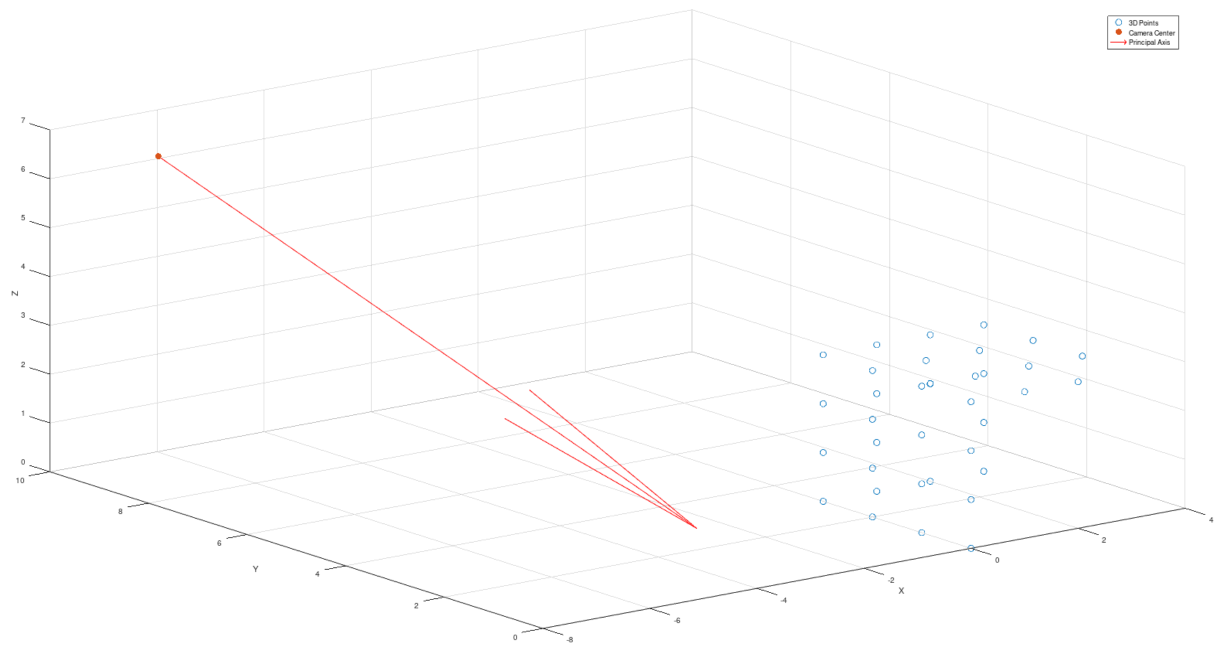

Figure 2: The 3D plot showing the camera center, the world points, & the principal axis

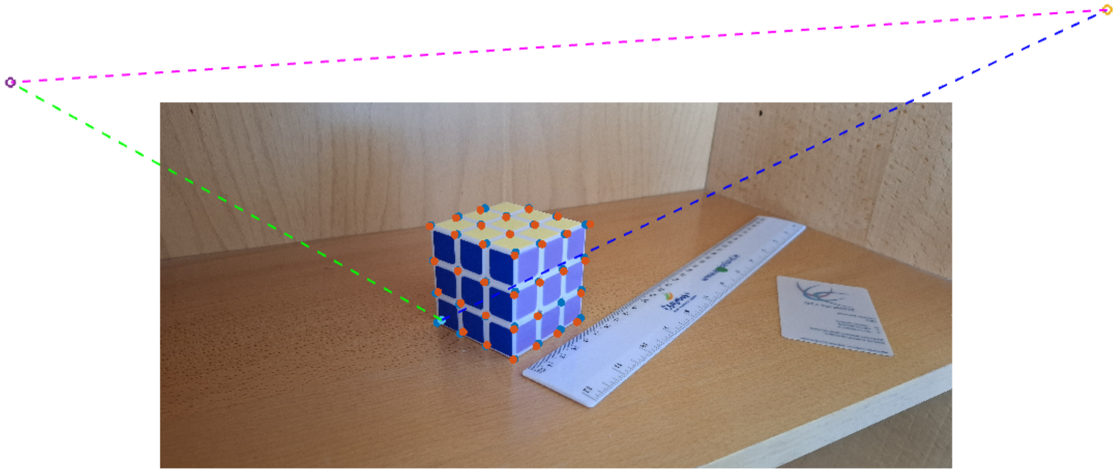

Figure 3: The image with projected 3D points & vanishing lines

2 Using Your Own Image from Your Camera for P-Matrix Estimation

```
filename = rubiks2.jpg
Click on the image to select 96 points and enter their 3D world coordina
tes.
Computed camera matrix P:
   1.8945e-01   1.1664e-01  -1.8014e-02   4.7870e-01
  -2.9295e-02   2.7453e-02  -2.1148e-01   8.2139e-01
   4.4202e-05  -4.2414e-05  -1.8274e-05   1.0123e-03
Intrinsic matrix K:
   3.3684e+03   1.2432e+01   9.1910e+02
            0   3.3496e+03   3.4387e+02
            0            0   1.0000e+00
Rotation matrix R:
  -6.9190e-01  -7.2199e-01   2.1206e-03
   2.0779e-01  -1.9632e-01   9.5827e-01
  -6.9144e-01   6.6347e-01   2.8586e-01
Camera center (in world coordinates):
   -9.0327
   11.5926
    6.6402
>> |
```

Figure 4: Command window output showing he computed camera matrix P, the intrinsic matrix K, & the rotation matrix R

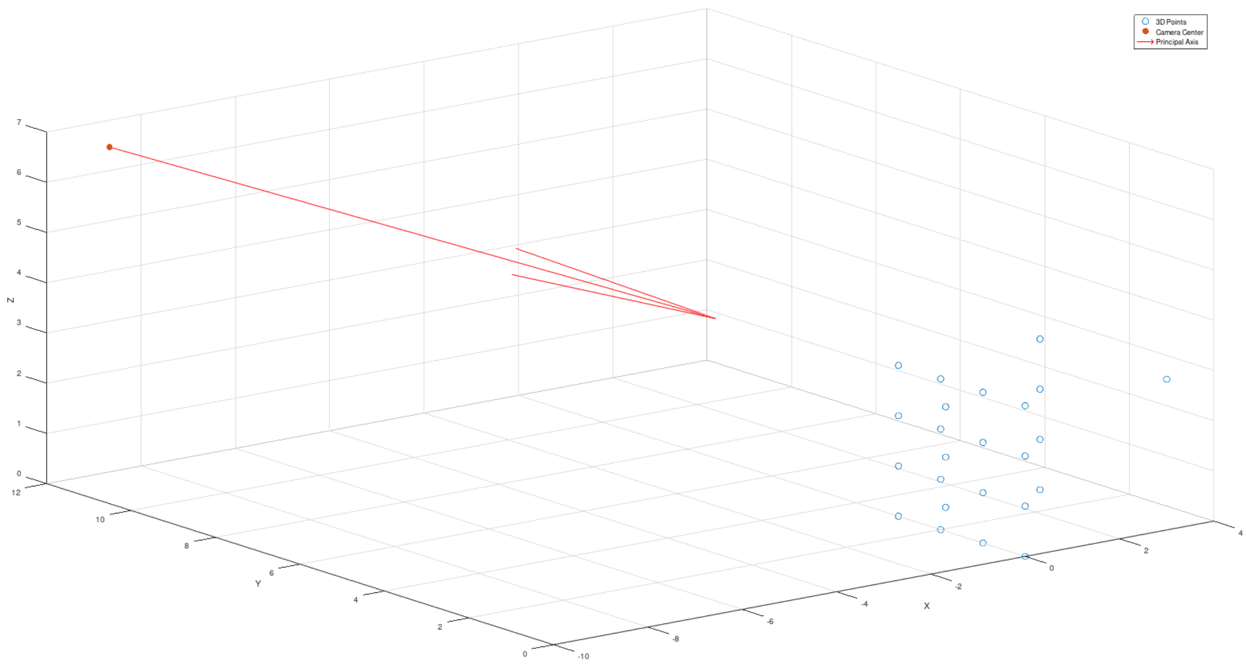

Figure 5: The 3D plot showing the camera center, the world points, & the principal axis

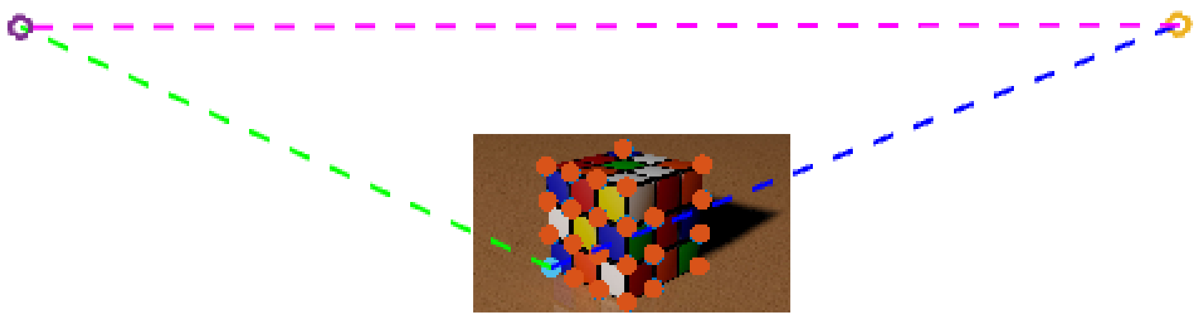

Figure 6: The image with projected 3D points & vanishing lines

3 Experiment & Reflect

3.1 How does increasing the number of points affect the accuracy & stability of the *P*-matrix estimation?

As the number of control points increased, the accuracy and stability of the estimated P Matrix improved. With 12 points, we observed discrepancies in the back-projected 3D points, while results with 40 points were far more consistent. The intrinsic and rotation matrices derived from the P Matrix appeared less sensitive to noise with more points, enhancing the reliability of the calibration.

3.2 Is there a noticeable difference in the accuracy of the back-projection when using fewer points versus more points?

Using fewer points (e.g., 12) resulted in higher deviations in back-projected points compared to their actual image locations. With 40 points, the back-projection closely matched the real-world setup, minimizing errors.

3.3 What challenges did you encounter when manually selecting points & entering 3D world coordinates?

The primary challenge that we faced when manually entering selecting the points was the precision: it was extremely difficult to precisely select the correct points due to the imprecision of the mouse as a selection device, human error, and a lack of fine-grain zoom control in the MATLAB UI.

We also found the process of manually entering the points very time-consuming and error-prone. If we mis-clicked a point or accidentally entered in the wrong world coordinate, it would greatly damage the accuracy of the entire calibration and we would be forced to start over again.