2D-3D Pose Estimation Mini-Project

Tomer Varsanno : 206727471
Naor Guetta : 316017359
Tamer Abu Shakra : 319147260
Aviv Azar : 209668144

August 23, 2024

Introduction

In this project, we implemented a 3D rendering system using OpenGL, featuring a teapot and a mountain terrain, each augmented with colored spheres. These spheres serve as reference points for applying the EPnP algorithm to estimate the camera's pose based on 2D-3D correspondences. The system offers two modes: a Debug mode for visualizing the user's movements and a User mode for interacting with the scene.

Implementation Details

Teapot and Terrain Scene

The teapot was rendered as a basic 3D model within the OpenGL environment, with pre-planted colored spheres distributed around it to facilitate pose estimation. Similarly, a mountain terrain was generated using a grid-based height map, with spheres placed strategically to simulate natural landmarks.

Debug and User Modes

In Debug mode, the system displays the user's trajectory and the distribution of reference points, allowing for a clear distinction between user input and system processing. User mode provides an interactive experience where the user can navigate the environment using the WASD keys and rotate the camera with arrow keys.

User Controls

The user can move the camera using the WASD keys and rotate the camera angle with the arrow keys—left and right for sideways rotation, and up and down for vertical movement. By pressing the 'B' key, the user can capture an image with at least four visible reference points, which is necessary for the EPnP algorithm to function correctly. Pressing 'R' processes all captured images to estimate the camera's position and overlays these estimations on the right-side user view. The user can navigate through these overlays using the left and right arrow keys.

Pose Estimation Process

Capture and Processing

Users capture images containing at least four colored reference points by pressing the 'B' key. These images are then processed using the EPnP algorithm when the 'R' key is pressed, resulting in an estimated camera pose that is overlaid on the right-side view for comparison.

Teapot Scene Results

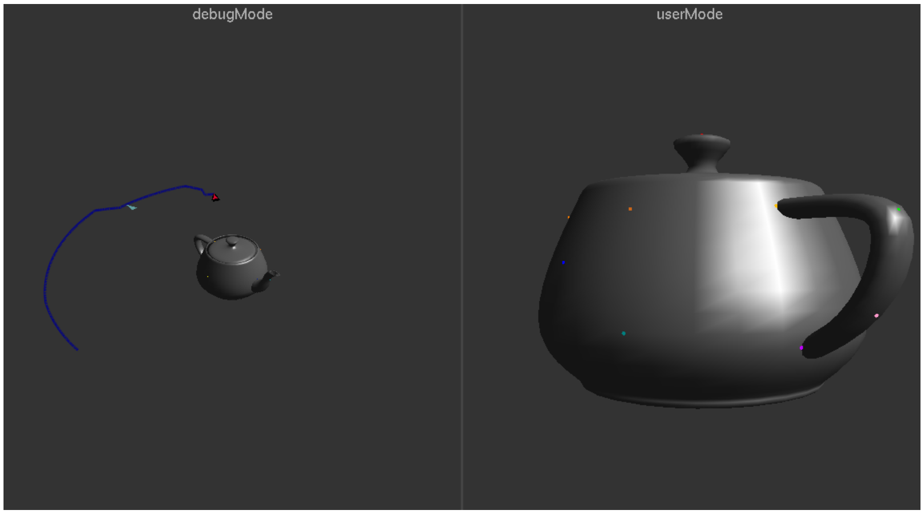
The images below illustrate the teapot scene before and after the EPnP overlay is applied.

Example 1:

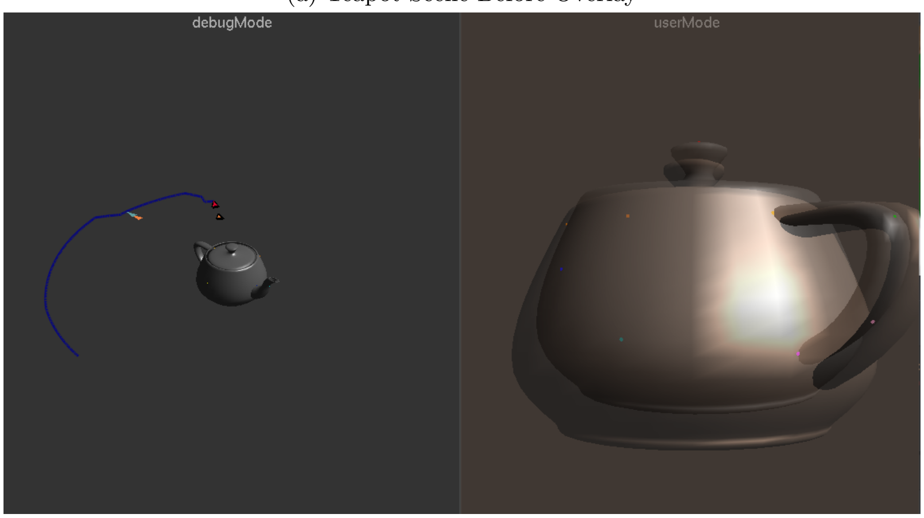
Figure 1: Teapot Scene Before Overlay

Figure 2: Teapot Scene After Overlay

Example 2:



(a) Teapot Scene Before Overlay



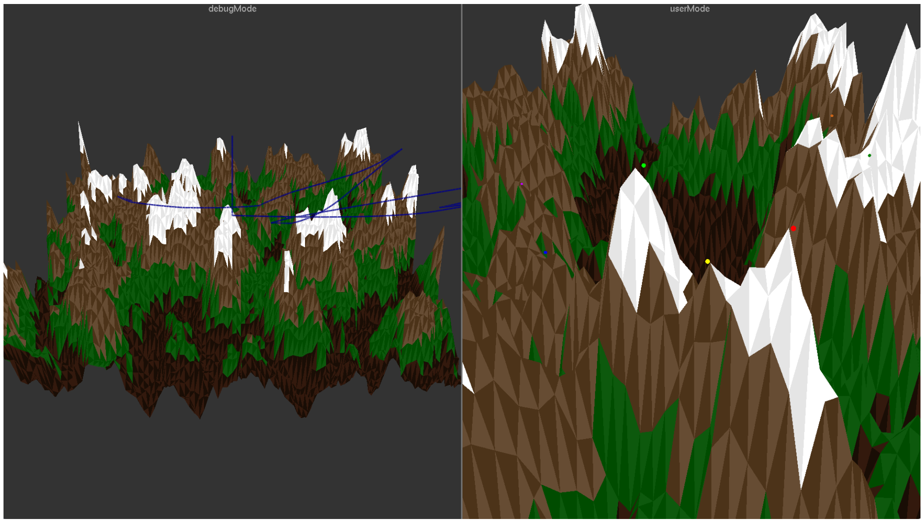
(b) Teapot Scene After Overlay

Figure 3: Comparison of Teapot Scene Before and After Overlay

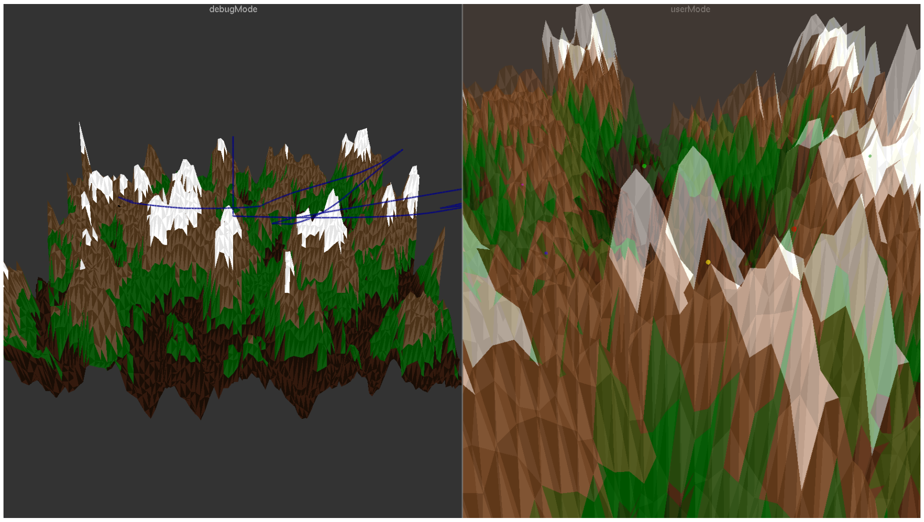
Mountain Terrain Results

Similarly, the following images show the results for the mountain terrain before and after the EPnP overlay.

Example 1:



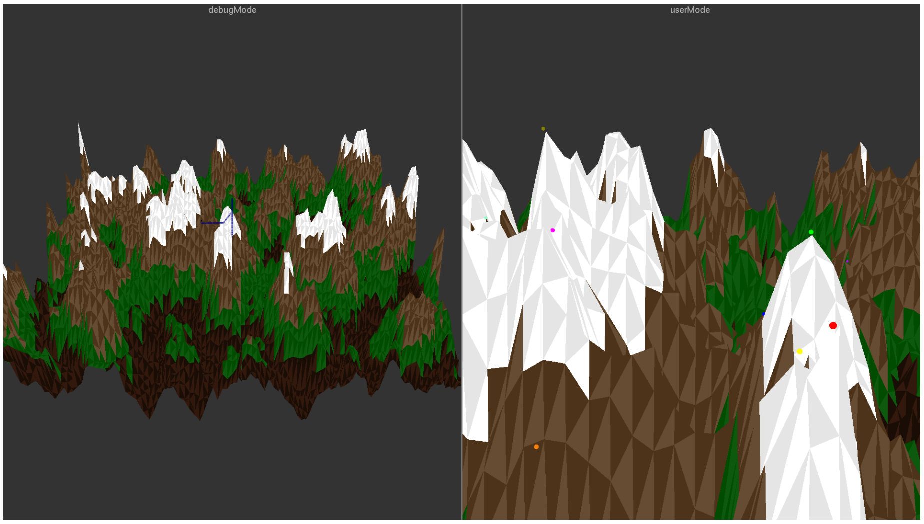
(a) Mountain Scene Before Overlay



(b) Mountain Scene After Overlay

Figure 4: Comparison of Mountain Scene Before and After Overlay

Example 2:



(a) Mountain Scene Before Overlay

(b) Mountain Scene After Overlay

Figure 5: Comparison of Mountain Scene Before and After Overlay

Error Calculation Methodology

In this project, we calculated the error margins for the estimated camera poses by comparing the estimated and actual positions and orientations. The errors are calculated as follows:

Positional Error (XYZ Coordinates)

The positional error measures the difference between the estimated and actual camera positions in 3D space. We compute this using the Euclidean distance between the estimated position $(x_{\text{est}}, y_{\text{est}}, z_{\text{est}})$ and the actual position $(x_{\text{actual}}, y_{\text{actual}}, z_{\text{actual}})$. The formula is:

$$\text{Positional Error} = \sqrt{(x_{\text{est}} - x_{\text{actual}})^2 + (y_{\text{est}} - y_{\text{actual}})^2 + (z_{\text{est}} - z_{\text{actual}})^2}$$

We also calculate the error in each individual coordinate (X, Y, Z) as:

$$\text{X Error} = |x_{\text{est}} - x_{\text{actual}}|$$

$$\text{Y Error} = |y_{\text{est}} - y_{\text{actual}}|$$

$$\text{Z Error} = |z_{\text{est}} - z_{\text{actual}}|$$

Angular Error (Yaw and Pitch)

The angular error quantifies the difference between the estimated and actual orientations of the camera. Specifically, we calculate the absolute difference between the estimated and actual yaw and pitch angles:

$$\text{Yaw Error} = |\text{Yaw}_{\text{est}} - \text{Yaw}_{\text{actual}}|$$

$$\text{Pitch Error} = |\text{Pitch}_{\text{est}} - \text{Pitch}_{\text{actual}}|$$

These calculations provide insights into both the accuracy of the camera's positional estimation and its orientation relative to the actual values.

Error Margin and Deviation Results

The following table summarizes the estimated and actual camera positions (XYZ coordinates) and orientations (Yaw and Pitch), along with the calculated errors.

Table 1: Error Analysis for Pose Estimation (Samples 1-4)

	Sample 1	**Sample 2**	**Sample 3**	**Sample 4**
x	44.078	9.37511	-23.0203	41.2522
y	30.5259	45.0205	68.1832	20.9894
z	90.675	108.597	112.978	43.6516
Yaw	-90.5	-64	-39.5	119
Pitch	-19	-29	-31.5	-26.5
Estimated x	44.1637	7.28444	-30.2408	45.2153
Estimated y	29.2634	49.6629	60.7617	25.9894
Estimated z	88.0719	105.084	115.377	45.7734
Estimated Yaw	-91.6089	-65.7881	-40.5328	118.989
Estimated Pitch	-18	-25.6448	-26.3641	-25.6687
x error^2	0.00734449	4.370901049	53.58972025	15.70616161
y error^2	1.59290625	12.341169	55.07866225	25
z error^2	6.77612961	6.185785949	5.755201	4.50203524
Positional Error	2.894370458	6.185785949	10.69689598	6.723704102
Yaw abs error	1.1089	1.7881	1.0328	0.011
Pitch abs error	1.0000	3.3552	5.1359	0.8313

Table 2: Error Analysis for Pose Estimation (Samples 5-8)

	Sample 5	**Sample 6**	**Sample 7**	**Sample 8**
x	46.8143	59.6307	4.05064	80.1797
y	34.7302	43.6474	29.3616	30.6261
z	96.9335	75.651	76.5404	40.0955
Yaw	-99.9218	-141.5	-20	-190.6
Pitch	-18.0301	-30	-23.5	-22
Estimated x	46.2725	62.0808	-0.872541	83.4271
Estimated y	32.6635	45.9688	25.3731	26.49344
Estimated z	102.11	67.7349	80.7239	39.3443
Estimated Yaw	-98.5	-140.263	-22.6226	-197.433
Estimated Pitch	-18.5	-26.7543	-19.8742	-19.7772
x error^2	0.29354724	6.00299001	24.23771116	10.54560676
y error^2	4.27124889	5.38889796	15.90805248	17.07887868
z error^2	26.79615225	62.66463921	17.50167225	0.56430144
Positional Error	5.600084676	8.605610215	7.592590855	5.309311337
Yaw abs error	1.4218	1.237	2.6226	6.833
Pitch abs error	0.4699	3.4257	3.6258	2.2228

Conclusion

In summary, this project successfully implemented a 3D rendering system capable of estimating camera poses using the EPnP algorithm. The system was tested with both a teapot and a mountain terrain scene, and the results demonstrated the accuracy and efficiency of the method. Future work could involve refining the error margins, exploring alternative pose estimation techniques, and extending the system to handle more complex environments.