This assignment aimed to create a pipeline revolving around Stereo vision, especially epipolar geometry, and 3D reconstruction. The goal is to extract meaningful information from stereo images captured by two virtual cameras. Using circle detection, correspondence matching, and 3D reconstruction algorithms, reconstruction of 3D locations and dimensions of spheres placed in the scene. Ultimately, investigating the behaviour of the Stereo vision model with various robust parameters and the performance of this model.

Through this process, it was discovered that:

- The performance of the stereo vision pipeline is heavily influenced by the position of the cameras and positions of the points of interests.
- Removal of noise is important during the corresponding point process for the 3D Reconstruction

This assignment demonstrates the integration of theoretical principles with practical implementation, offering a hands-on understanding of stereo vision and 3D reconstruction challenges.

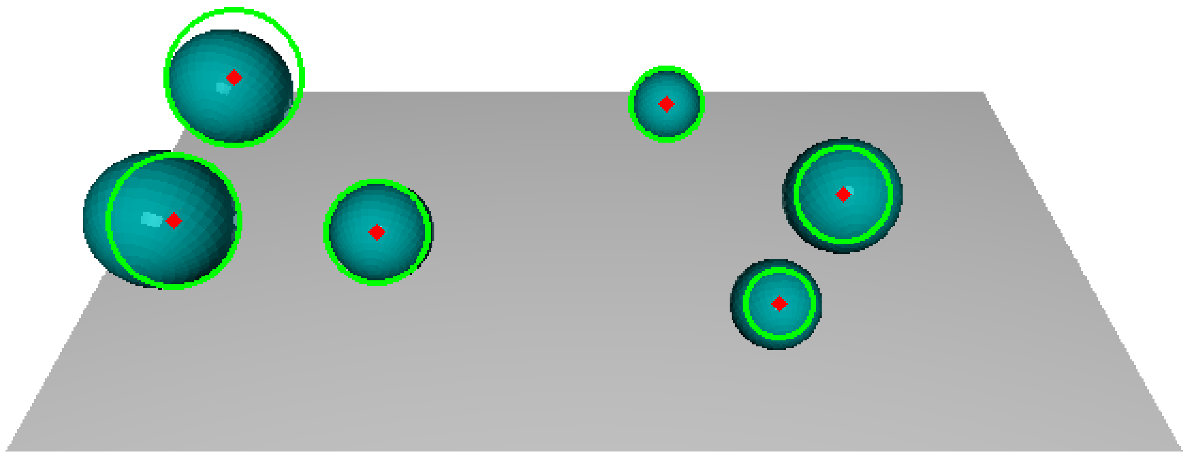

*Figure 1: View1.png image from camera0's Circle detection result. Missing detections due to huge overlapping occuring.*

Hough Circle Transform provide detection of the spheres, providing its 2D world coordinates and its radius. With spheres overlapping due to random sizing and position, minimum distance parameter is important to adjust to increase overall accuracy of the model.

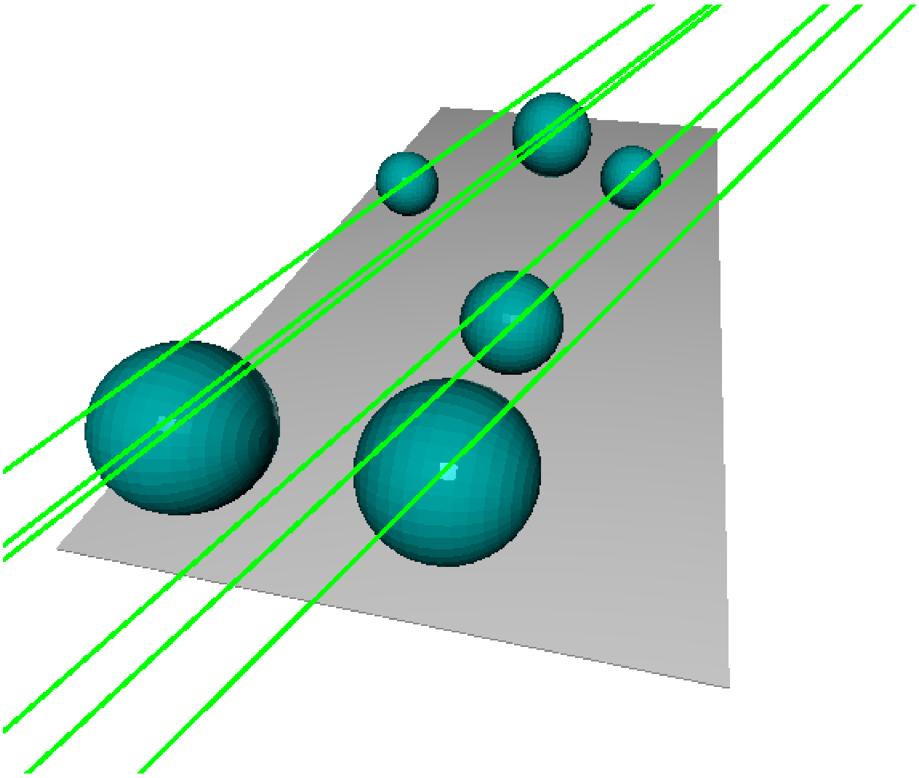

*Figure 2: Task 4 View1.png image from camera1, resullt of Epipolar Lines drawn.*

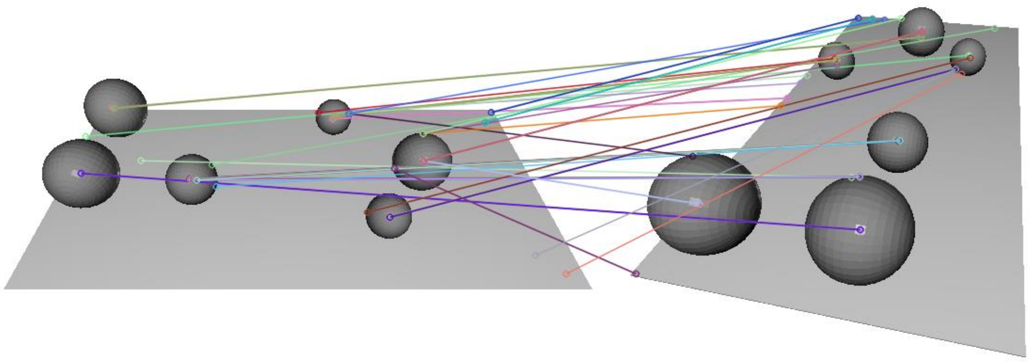

*Figure 3: Task 5 Instance of correspondence matching with some non matching done with the platform and the spheres.*

Used SIFT to detect and describe distinctive features in both images. In addition, used Fundamental Matrix,F, to get the coordinates on the epipolar line to compare the difference between the keypoint detected by the SIFT algorithm and the point on the epipolar line, and whether its within the threshold to be considered. However, additional non-essential features such as the corners of the platforms are being considered, affecting the next few tasks. Reducing threshold will produce stricter matching process. Post cleaning would be needed for 3D Reconstruction.

For task 6, used the 3D Reconstruction Algorithm as discussed in lecture, for each matched points:

(1) $ap_L - bR^T p_R - T - c(p_L \otimes R^T p_R) = 0$

(2) $H \begin{pmatrix} a \\ b \\ c \end{pmatrix} = T$

(3) $\hat{P} = (ap_L + bR^T p_R + T)/2$

where $\hat{P}$, is the 3D coordinate.

Evaluating the performance of the 3D Reconstruction,

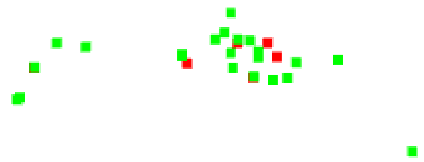

*Figure 4: 3D Reconstruction points (green) with ground truth (red)*

With additional features being considered, there are more matches than the number of spheres on the plane. This is because of the SIFT process. With this, an algorithm is needed to pick the nearest green with the nearest red to accurately dictate how far center of spheres are from the relevant 3D points, which is the defined as the measurement of error. The error statistics are as follows:

PAIR [-10.   1.6   1. ] and [-0.7855951  -7.89706634  3.73100262]: Error = 13.5114 units

PAIR [-5.   1.2  1. ] and [ 0.95581759 -6.65515435  0.43556841]: Error = 9.8739 units

PAIR [6.  1.4 0. ] and [ 0.975403   -6.56200884 -0.47198881]: Error = 9.4267 units

PAIR [-10.   1.6  -4. ] and [-0.05916    -5.12246736 -8.05633348]: Error = 12.6675 units

PAIR [ 2.  1. -4.] and [ 0.98907828 -6.51644996 -0.94393698]: Error = 8.1767 units

PAIR [4. 1. 3.] and [ 2.18536887 -6.97057364  2.75913278]: Error = 8.1781 units

Mean Error: 10.3057 units

Max Error: 13.5114 units

To find the radii of Sphere Construction based on 3D Reconstructed points,

$$Radii_{ref} = Radius_{ref} * \ Depth_{ref}/focal\_length$$

$$Radii_{view} = Radius_{view} * \ Depth_{view}/focal\_length$$

$$Radii_{ave} = (Radius_{ref} + Radius_{view})/2$$

Sphere 1: Estimated Radius = 0.3731 units

Sphere 2: Estimated Radius = 0.0352 units

Sphere 3: Estimated Radius = -0.0466 units

Sphere 4: Estimated Radius = -0.5727 units

Sphere 5: Estimated Radius = -0.0523 units

Sphere 6: Estimated Radius = 0.1230 units