1. Point Cloud Reorientation

Task: Given a 3D point cloud of an object that includes scene floor as the background, that task is to re-orient the point cloud, such that the floor is aligned with $x = 0$ plane.

Approach:

1. Find the equation of the plane passing through the scene floor.
2. Bring the center of the floor to the origin and find the rotation matrix.
3. Rotate the whole point cloud with the computed rotation matrix

Equation of the plane: The plane equation is shown in Eq. 1. The coefficients (a, b, c) give the direction of the plane's normal vector, and d tells the perpendicular distance to the plane from the origin.

$$ax + by + cz = d \tag{1}$$

The first step towards solving this task is to find the equation of the plane passing through the scene floor. However, this is not a trivial task, as there can be multiple planar surfaces existing in the given scene. The assumption made in this assignment is that the floor is the most dominant planar entity present in the scene.

We use two implementations to find the plane equation passing through the floor. The first is Open3D's plane segmentation method, and the second is a simple implementation of the RANSAC method from scratch. We now discuss the RANSAC-based approach to find the equation of the biggest plane. The idea is to pick a random plane, find its inliers, and keep doing this in the hope that we will find the plane with the most number of inliers. For a given the scene points cloud $P = \{p_1, p_2, \ldots p_n\}$, having n points, randomly pick 3 points ($p_1^{'}$, $p_2^{'}$, $p_3^{'}$). Get the vectors $v1$ and $v2$ by, $v1 = p_2^{'} - p_1^{'}$ and $v2 = p_3^{'} - p_1^{'}$. Then, we find the vector $v3$ normal to these two vectors by taking a cross-product as $v3 = v1 \times v2$. The values of $v3$ are the coefficients of the plane (a, b, c) respectively. We find the coefficient d by, (a, b, c) $\cdot$ $p_2^{'}$, where $\cdot$ denotes vector dot-product. Now that we have the equation of the plane on which ($p_1^{'}$, $p_2^{'}$, $p_3^{'}$) lie, we compute the distance of all the points in the scene to the plane. Then, we find all the points that are less than a pre-defined threshold distance th that are the inliers. We repeat the process for $iters$ times and record the plane equation that has the biggest set of inliers. We use $th = 0.01$ and $iters = 1000$ in our experiments. Since we assume that the scene floor is the biggest plane, we get its plane equation.

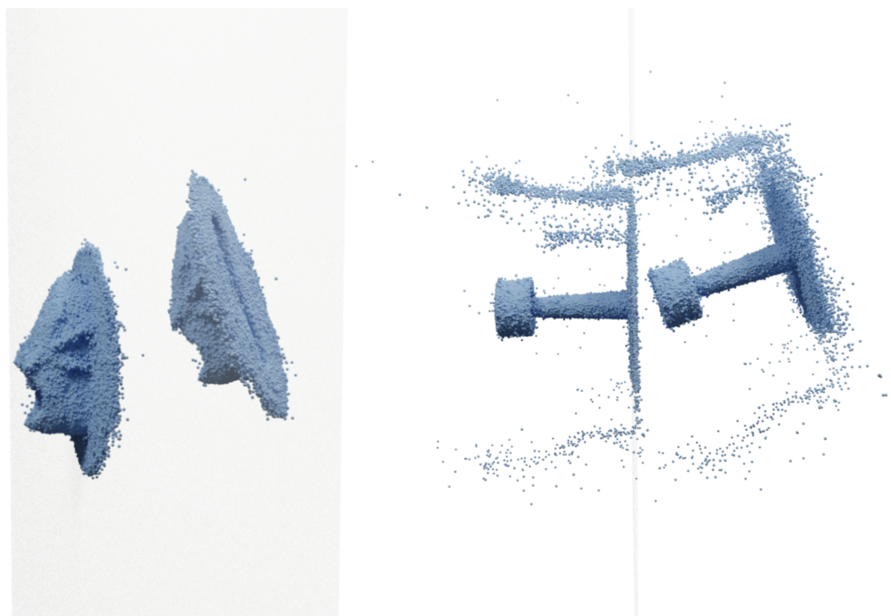

Figure 1: Examples of point cloud reorientation, where the floor is aligned with $x = 0$ plane. Left is the aligned and right is the original point cloud.

Aligning the floor: If we want to align the floor to $x = 0$ plane, the normal of the target plane is $(1, 0, 0)$. We use the Rodrigues Formula given in Eq. 2 to find the rotation matrix to align the normal of the two planes.

$$\mathtt{R} = v\cos(\theta) + (k \times v)\sin(\theta) + k(k \cdot v)(1 - \cos(\theta)) \tag{2}$$

Here, k is the unit vector describing an axis of rotation about which v rotates by an angle θ.

Rotating the scene: Once we find the rotation matrix to align the scene floor with the desired axis plane, we can apply the rotation transformation on the scene point cloud P. The rotated point cloud can be found by $\hat{P} = \mathtt{R} \cdot P$. Some examples are shown in Figure 1

Limitations: This method would fail when the assumption does not hold true. If there are multiple planes existing in the scene point cloud that are bigger than the floor, the algorithm will fail to find the plane fitting the floor. This is also observed in Figure 2.

Figure 2: Failure case, where the floor does not align with the $x = 0$ plane. Right is the aligned and left is the original point cloud.

2. Surface Reconstruction

Task: Estimate the surface of the given point cloud.

Approach:

1. Downsample the given point cloud.
2. Remove outliers from the scene.
3. Use Poisson Reconstruction to get the surface.

By observing the given point clouds, we find that there exist some noisy points both in the form of outliers and on the surface. To get a decent Poisson surface reconstruction, we first downsample the scene point cloud using the voxel downsampling method. This reduces the surface noise to some extent. We then remove the outliers using the ball radius method. These pre-processing steps give decent surface reconstruction, as shown in Figure 3. Failure case is shown in Figure 4. To improve the surface reconstruction, we can use more advanced methods like finding the signed distance field representation and then extracting the mesh using Marching Cubes.

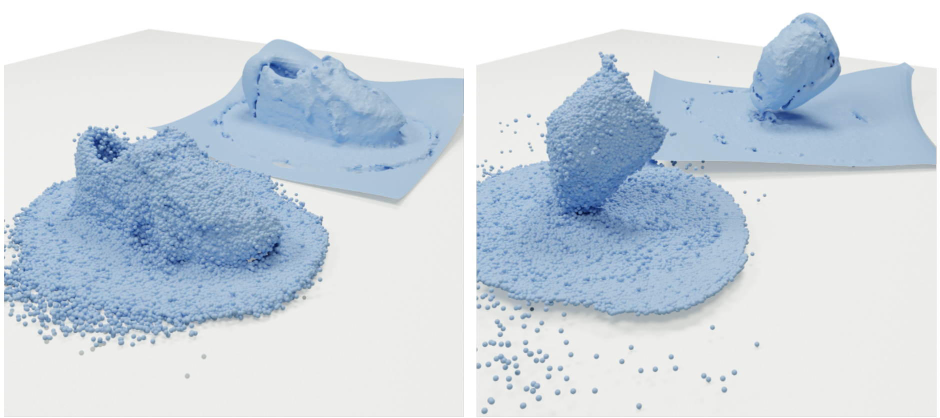

Figure 3: Examples of Poisson surface reconstruction.

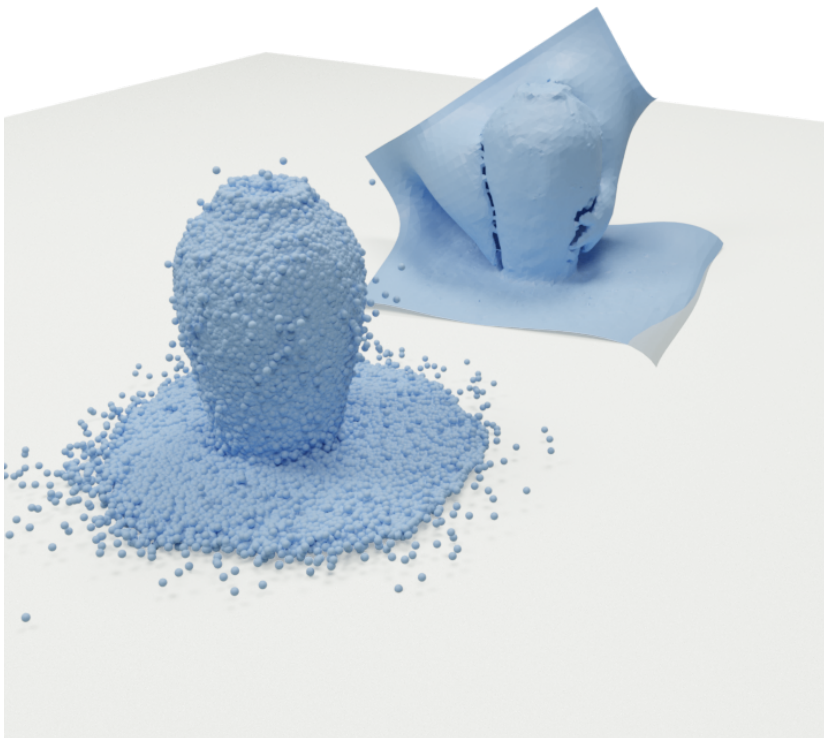

Figure 4: Faliure case by Poisson surface reconstruction.