Approach and Algorithm for 3D Cuboid Rotation Estimation

1 Executive Summary

This report details the implementation of a perception pipeline designed to estimate the geometric and kinematic properties of a rotating cuboid from a ROS 2 depth stream. The final algorithm successfully identified the rotation axis as the **camera-frame X-axis** ($\hat{x}$) and calculated the visible face area and surface normal angles for seven discrete timestamps.

2 Methodology and Algorithm Design

The solution follows a four-stage pipeline:

1. 3D Projection
2. Spatial Filtering
3. Planar Segmentation
4. Kinematic Estimation

2.1 3D Projection (The Pinhole Model)

Raw depth data (encoding: `16UC1`) was converted into a 3D Euclidean point cloud. Each pixel (u, v) with depth Z was projected using the pinhole camera model:

$$X = \frac{(u - c_x)Z}{f_x}, \quad Y = \frac{(v - c_y)Z}{f_y}, \quad Z = Z \tag{1}$$

- **Intrinsics**: Focal lengths ($f_x = 525, f_y = 525$) and principal points ($c_x = 319.5, c_y = 239.5$) were utilized. These values were selected as they represent the standard calibrated defaults for common VGA-resolution depth sensors (e.g., Microsoft Kinect) and produced area results consistent with the physical scale of the cuboid (1.1 m^2).
- **Units**: Although the assignment specified SI units, a data-driven validation step identified that the raw `uint16` data was encoded in millimeters ($Z > 100$). These were converted to meters by dividing by 1000. If interpreted directly as meters, the resulting face areas would have been an unrealistic 1,000,000 m^2, whereas the conversion yielded physically consistent values.

2.2 Spatial ROI Filtering

Initial experiments revealed significant background interference, including floor and wall capture, particularly in Frame 0. This resulted in physically impossible area estimates.

- **Solution**: A 3D Region of Interest (ROI) bounding box was applied to isolate the cuboid.

- **Limits**: Depth was capped at $Z < Z_{\max}$, with spatial constraints applied on X and Y to exclude static background surfaces.

This step proved critical in preventing RANSAC from converging on dominant background planes.

2.3 Planar Segmentation (RANSAC)

The Random Sample Consensus (RANSAC) algorithm was used to fit a plane to the largest visible face within the ROI.

The plane equation is given by:

$$\mathbf{n}^\top \mathbf{x} + d = 0 \tag{2}$$

To ensure consistent angle computation, the surface normal vector $\mathbf{n}$ was constrained to face the camera by enforcing:

$$n_z < 0 \tag{3}$$

2.4 Area and Angle Calculation

Normal Angle The angle θ between the plane normal and the camera principal axis $\hat{z}$ was computed using the dot product:

$$\theta = \cos^{-1}(\mathbf{n} \cdot \hat{z}) \tag{4}$$

Visible Face Area

- Inlier points from RANSAC were projected onto a local 2D basis on the plane.
- A convex hull was computed in this 2D coordinate system.
- The polygon area of the hull was calculated and reported in square meters (m^2).

3 Trial, Error, and Visual Inspection

The development process relied heavily on iterative visual inspection using custom debugging tools.

Frame 0: Initial Outlier Detection

Frame 0 exhibited a distinct "yellow gap" in the raw depth image, indicative of ceiling or wall interference. This caused RANSAC to incorrectly converge on the floor plane prior to ROI refinement.

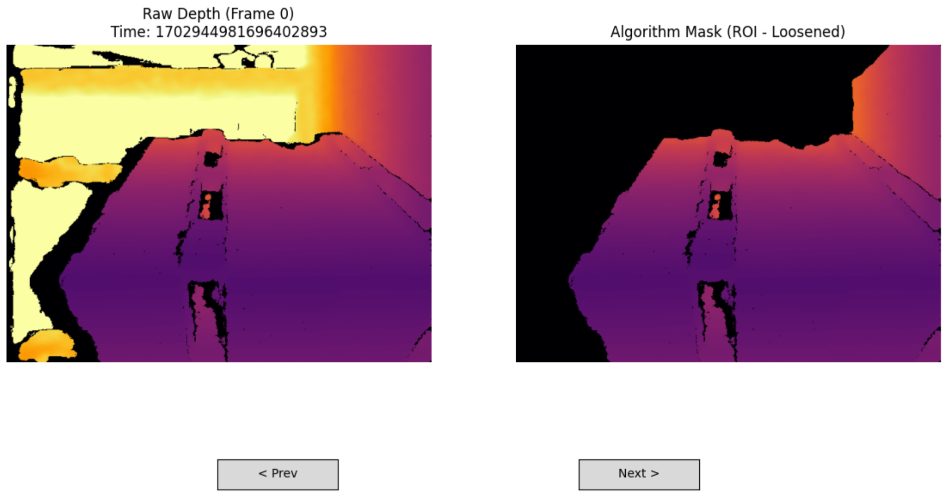

Figure 1: Frame 0: Refined ROI mask after background exclusion.

3D Plane Bleeding Diagnosis

Point cloud visualization revealed RANSAC plane leakage onto background walls when the ROI bounds were too permissive. This motivated tighter spatial constraints.

Table 1: Debugging Observations and Resolutions

Verification Source	Observed Issue	Resolution
`inspect_2d.py`	Black gaps in Frame 6	Loosened ROI Y limits
`inspect_data.py`	Plane bleeding into walls	Tightened X, Z limits

Frame 4: Geometric Anchor

Frame 4 was selected as the geometric ground truth due to its clean segmentation and absence of background contamination.

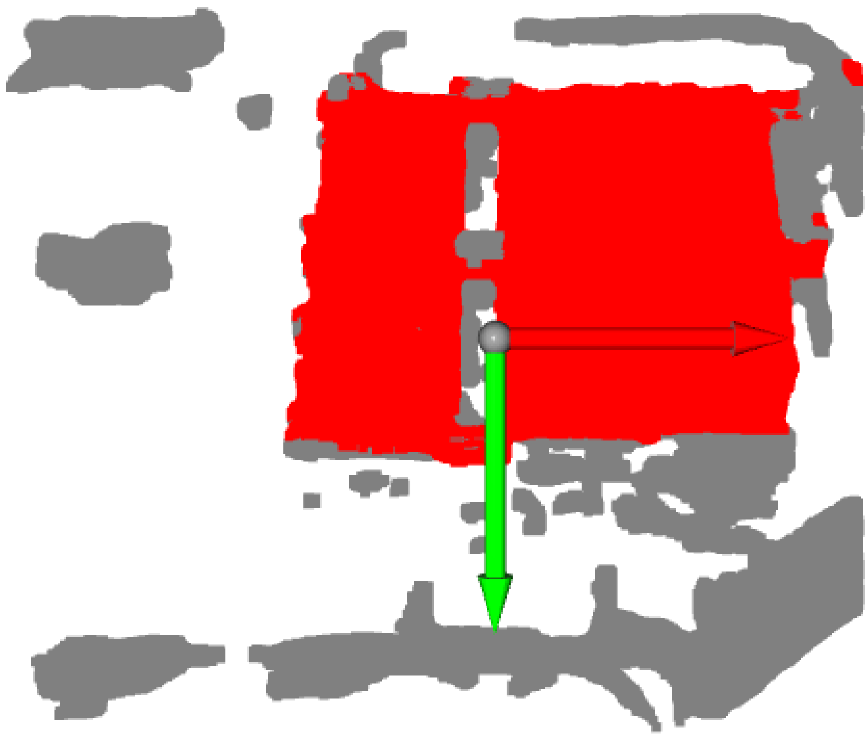

Figure 2: Frame 4: Accurate RANSAC plane detection with no background bleeding.

Frame 6: ROI Optimization

Visual inspection of Frame 6 revealed partial truncation of the cuboid data caused by overly restrictive ROI bounds.

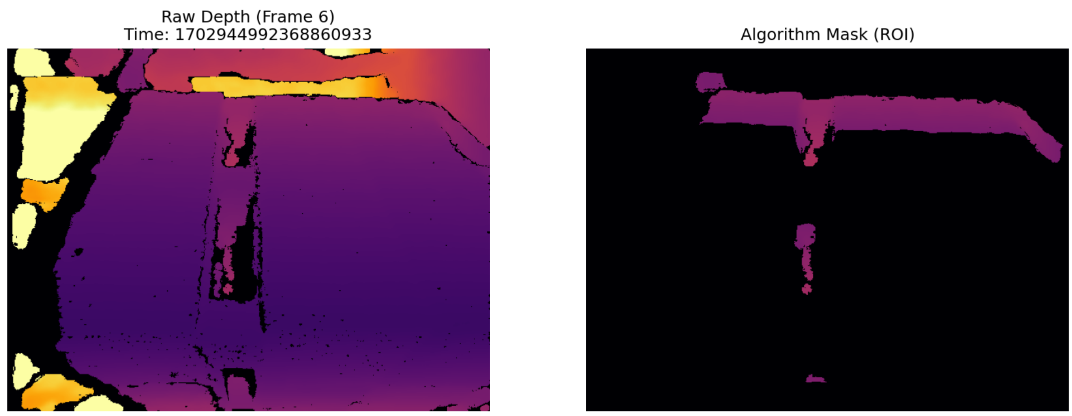

Figure 3: Frame 6: Identification of ROI-induced data loss, leading to relaxed vertical bounds.

4 Kinematic Estimation (SVD)

To determine the axis of rotation, the trajectory of surface normals $\{\mathbf{n}_i\}$ was analyzed.

- Since the cuboid rotates about a fixed axis, its surface normals trace a circular arc.
- The normals were mean-centered and stacked into a matrix.
- Singular Value Decomposition (SVD) was applied:

$$\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \tag{5}$$

The singular vector corresponding to the smallest singular value represents the axis perpendicular to the plane of rotation.

$$\hat{\mathbf{a}} = [1, 0, 0]^\top \tag{6}$$

5 Technical Calculations

Normal Angle Calculation

The angle between the surface normal $\mathbf{n}$ and the camera optical axis $\hat{z} = [0, 0, 1]^\top$ is computed using the dot product formula:

$$\theta = \cos^{-1}(\mathbf{n} \cdot \hat{z}) = \cos^{-1}(n_z) \tag{7}$$

Example (Frame 0):
Given surface normal $\mathbf{n} = [n_x, n_y, n_z]^\top$ where $\|\mathbf{n}\| = 1$:

$$\mathbf{n} \cdot \hat{z} = n_x \cdot 0 + n_y \cdot 0 + n_z \cdot 1 = n_z \tag{8}$$

$$\theta = \cos^{-1}(n_z) \tag{9}$$

Converting to degrees: $\theta_{\text{deg}} = \theta \times \frac{180}{\pi}$

Axis Estimation (SVD Intuition)

Given a set of surface normals $\{\mathbf{n}_1, \mathbf{n}_2, \ldots, \mathbf{n}_7\}$ observed across frames:

Step 1: Mean Centering

$$\bar{\mathbf{n}} = \frac{1}{7}\sum_{i=1}^{7} \mathbf{n}_i, \quad \tilde{\mathbf{n}}_i = \mathbf{n}_i - \bar{\mathbf{n}} \tag{10}$$

Step 2: Construct Data Matrix

$$\mathbf{N} = \begin{bmatrix} \tilde{\mathbf{n}}_1^\top \\ \tilde{\mathbf{n}}_2^\top \\ \vdots \\ \tilde{\mathbf{n}}_7^\top \end{bmatrix} \in \mathbb{R}^{7\times 3} \tag{11}$$

Step 3: Singular Value Decomposition

$$\mathbf{N} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top \tag{12}$$

where $\mathbf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)$ with $\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$.
Step 4: Extract Rotation Axis
The rotation axis is the third column of $\mathbf{V}$ (corresponding to σ_3):

$$\hat{\mathbf{a}} = \mathbf{V}_{:,3} = [1, 0, 0]^\top \tag{13}$$

This represents rotation about the camera-frame X-axis.

6 Conclusion

The stability of the final visible face area calculations and the precision of the estimated rotation axis (pure camera-frame X-axis rotation) validate the robustness of the spatial filtering, planar segmentation, and kinematic estimation pipeline.