Edge and Corner Detection in Unorganized Point Clouds

for Robotic Pick and Place Applications

Isha Jangir, Siddharth Nimbalkar, & Rajat Kumar Thakur

Under the guidance of Dr. Deepika Gupta
Indian Institute of Information Technology Vadodara

April 23, 2025

Overview

Motivation

Preview:

- Over the past decade, the construction industry has struggled to improve its productivity, while the manufacturing industry has experienced a dramatic productivity increase.
- A possible reason is the lack of advanced automation in construction.
- For developing a system for such a task, the system must estimate the pose of the cartons in a clutter, grasp it, and arrange it in the appropriate stack for further processing.

In this project, we focus on estimating the pose of the objects in clutter, as shown. We assume that all the objects present in the clutter have the same dimensions, which is very common in warehouse industries.

Figure: Cartons Clutter

Problem Statement

Input: An unorganized 3D point-cloud of a scene containing multiple identical, textureless objects (e.g. bricks or cartons) in arbitrary clutter.

Objectives:

1. Identify edge points and their connecting line segments.
2. Locate corner intersections.
3. Estimate pose for each known object instance.

Key Challenges:

1. Noisy & unstructured data
2. Textureless surfaces
3. Real-time constraints

A reliable edge and corner detector enables autonomous robots to perceive and grasp objects in real-world industrial settings—without expensive preprocessing or extensive training data.

Related Work

While all the methods in this area have shown excellent results, they cannot be used directly for this type of work.

- Since the objects are textureless. Therefore the number of features will be less, making the feature matching methods less reliable.
- The performance of CNN-based algorithms relies on extensive training on large datasets. However, due to the large variety of objects in the warehouse industry it is challenging to train CNN for all types of objects.
- In (Bazazian et al., 2015), the author performs fast computation of edges by constructing covariance matrices from local neighbors, but the author has demonstrated for synthetic data only.
- In (Vohra et al., 2021), the authors proposed an edge and corner detection framework for unorganized point clouds targeting robotic pick and place applications, but their method involved normal estimation and clustering steps, making it computationally expensive, especially for large and noisy point clouds.

Model Workflow

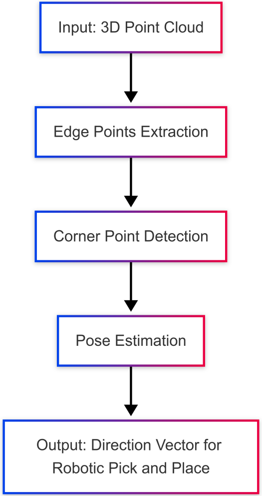


Figure: Workflow Diagram

Edge Points Extraction

We adopted the eigenvalue-based surface variation method from Bazazian et al. (2015) to efficiently extract sharp edge points, as it offers a faster alternative to other estimation methods.

Overview:

1. For each point, compute its 3×3 covariance matrix using its k-nearest neighbors
2. Obtain eigenvalues $\lambda_0 \leq \lambda_1 \leq \lambda_2$ of the covariance matrix.
3. Calculate surface variation: $\sigma_k(p) = \frac{\lambda_0}{\lambda_0+\lambda_1+\lambda_2}$
4. $$\sigma_k(p) = \begin{cases} 0, & \text{if } \lambda_0 \approx 0 \text{ (flat surface)} \\ \text{higher value}, & \text{if edge is present} \end{cases}$$
5. As the smallest eigenvalue of covariance matrix for the flat surfaces is zero then the value of the surface variation for the flat surfaces would be zero.

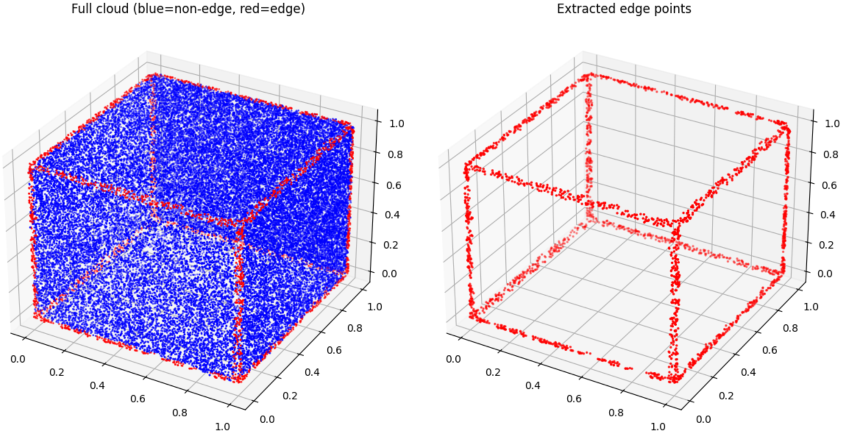


Figure: 3D Cube Edge Cloud

Corner Detection using Harris Corner Detection

Harris Corner Detector detects salient corner points from the extracted edge point cloud.

Overview:

1. Build a KDTree for fast nearest neighbor queries
2. For each point:
   - Find k-nearest neighbors
   - Compute local covariance matrix
   - Estimate normal vector
3. For each point:
   - Find neighbors within a given radius.
   - Compute difference of normals.
   - Calculate Harris response:

     $$R = \det(H) - k \times (\mathrm{trace}(H))^2$$

     where H is the covariance of normal differences
4. Apply:
   - Thresholding on R
   - Non-maximum suppression within a local radius.

Corner Points Extraction

From the detected corners, we select points with the minimum and maximum values along the X, Y, and Z axes. These extreme points serve as key references for further alignment or processing steps.

Advantages:

- Robust to noise in unorganized point clouds
- No surface reconstruction needed

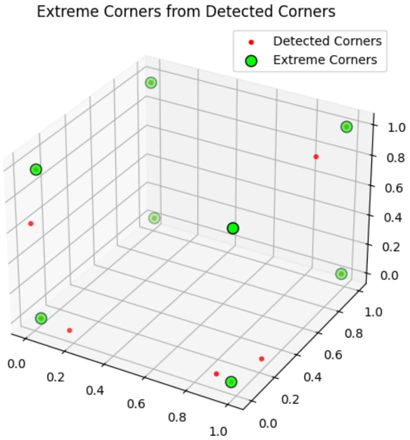


Figure: Corners Extracted from point cloud

Pose Estimation

We adopt this method from **Vohra et al. (2021)** for robust and precise pose estimation in unorganized point clouds.

Approach:

- Once the extreme corner points are identified, we estimate their corresponding positions in the local object frame.
- The directions of the principal axes (d1, d2) must be assigned carefully:
  - Determine whether each axis corresponds to $+x, -x, +y, -y, +z, -z$ in the object's local frame.
  - Assign the direction of d_1 and d_2 such that the third orthogonal axis points toward the camera.
- This approach ensures consistent and correct alignment of the object in 3D space.

Pose Estimation our on point cloud data

The estimated pose of the object in the camera frame is given by

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \qquad t = \begin{bmatrix} 0.5 \\ -0.5 \\ 1.5 \end{bmatrix}.$$

Here,

- R is the 3×3 rotation matrix aligning the object's local axes to the camera axes (in this case a 180° flip about the x–axis).
- t is the translation vector (in meters) from the camera origin to the object's local origin.

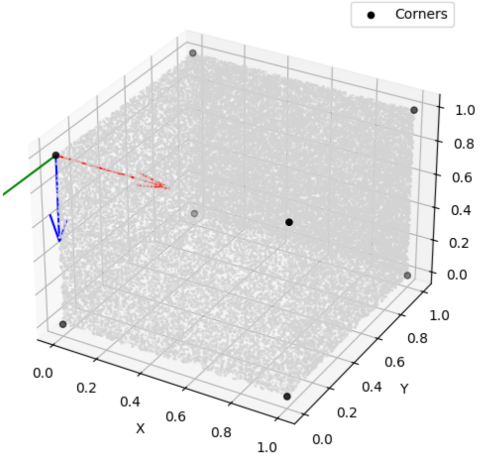


Figure: Estimated Pose

Results on Different Objects

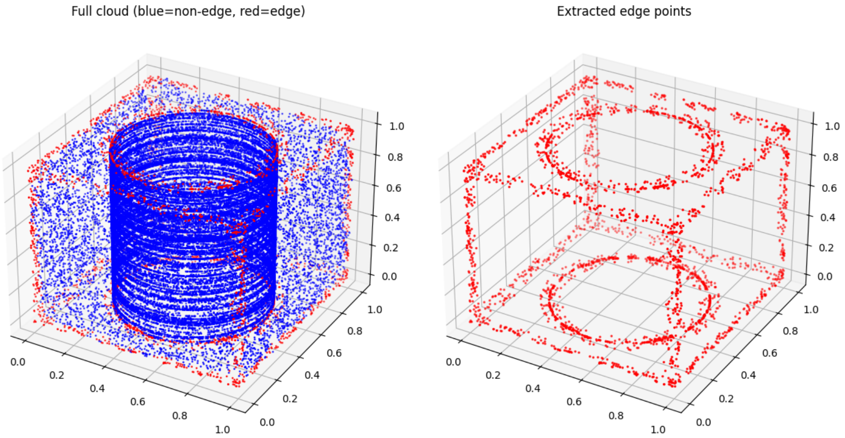
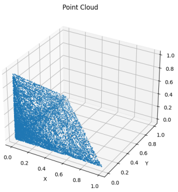
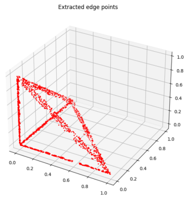


Figure: Hollow Cylinder inside a Cube

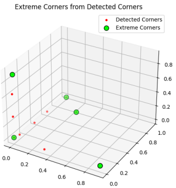
Results on Different Objects



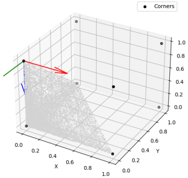
(a) Point Cloud of a Tetrahedron



(b) Edge Points of a Tetrahedron



(c) Corner Points of a Tetrahedron



(d) Pose Estimation of a Tetrahedron

Figure: Our model on a Tetrahedron

Experimental Results

To quantitatively compare runtime performance, we ran both our proposed pipeline and the Vohra et al. (2021) algorithm on the identical cube point cloud. Table "Computation time (sec) at each step" breaks down each stage.

Table: Computation time (sec) at each step

	Edges Points	**Corner Points**	**Pose Estimation**	**Total Time**
Our Model	5.520	0.573	1.850	7.943
Vohra (2021)	24.168	1.110	1.551	26.829

This clear speed-up—driven primarily by our eigenvalue-based, clustering-free edge extraction—demonstrates the computational efficiency of our approach on the same input data.

Conclusions & Referemces

A novel, fully-integrated edge-and-corner detection for unorganized point clouds was developed by combining a clustering-free eigenvalue-based edge extractor with a Harris-3D corner detector and a pose solver based on corners. Overall, this combined framework delivers fast, accurate 6D pose estimates suitable for real-time robotic pick-and-place applications.

References:

- N. Bazazian, X. Wang & J. Davis, "Fast & Robust Edge Extraction in Unorganized Point Clouds," in *Proc. IEEE Int. Conf. on Computer Vision*, 2015.
- M. Vohra, R. Prakash & L. Behera, "Edge and Corner Detection in Unorganized Point Clouds for Robotic Pick and Place Applications," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1733–1740, 2021.

The End