Edge and Corner Detection in Unorganized Point Clouds for Robotic Pick and Place Applications

Rajat Kumar Thakur
Department of Computer Science & Engineering
Indian Institute of Information Technology Vadodara
202211070@diu.iiitvadodara.ac.in

Isha Jangir
Department of Computer Science & Engineering
Indian Institute of Information Technology Vadodara
202211031@diu.iiitvadodara.ac.in

Siddharth Nimbalkar
Department of Computer Science & Engineering
Indian Institute of Information Technology Vadodara
202211057@diu.iiitvadodara.ac.in

Dr. Deepika Gupta
Department of Computer Science & Engineering
Indian Institute of Information Technology Vadodara
deepika_gupta@diu.iiitvadodara.ac.in

***Abstract*—In this paper, we present a fully integrated methodology for fast and robust edge and corner detection in unorganized 3D point clouds. Our method begins with an eigenvalue-based surface variation metric to efficiently extract sharp edge points from raw point cloud data. We then apply a 3D Harris corner detector on the extracted edge cloud to localize salient corner points by analyzing local normal variations, followed by non-maximum suppression to ensure spatial distinctiveness. Extreme corner points along the principal axes are used to compute consistent pose estimates for known, textureless objects. We then validate our approach on synthetic shapes to demonstrate the proposed technique's efficacy. We compare our method to the other solutions for 3D edge and corner extractions in unorganized point cloud data. We observed that the proposed method achieved less computation time than other existing algorithms for the same data cloud with minimal parameter tuning. The overall approach should be suitable for direct application to autonomous robotic grasping in cluttered warehouse environments.**

***Index Terms*—Edge Extraction, Unorganized Point Cloud, Harris Corner Detector**

I. Introduction

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. One of the main challenges in the automation of the above work is that the system has to deal with a stack of objects, which can come in random configurations, as shown in Fig. 1.

Most of the existing research on extracting edges in point clouds are considering either statistical and geometrical methods or estimating the normals on sharp edges. The challenge for estimating normals on the edge feature points are related to the neighbourhood employed for the normal estimation. The neighborhood may enclose points belonging to different surface patches across the edge feature. For developing a system for such a task, the system must estimate the pose of the cartons in a clutter, grasp them, and arrange them in the appropriate stack for further processing.

In this paper, we focus on estimating the pose of the objects in the clutter, as shown in Fig. 1. We assume that all the objects present in the Clutter have the same dimensions, which is very common in warehouse industries. The main contribution of this paper is to extract corner points for some 3D shapes. We intend to extract corners through Harris corner detection algorithm and then quantitatively compare the computational loads across various corner detection algorithms.

Fig. 1. Cartons Clutter (Image Courtesy: Internet)

The remainder of the article is organized as follows. Section II presents related work, followed by a description of our approach and architecture in Section III. Section IV reports the experimental results of our approach, and conclusions are drawn in Section V.

II. Related Work

While methods have shown excellent results for edge or corner detection, they cannot be used directly for our work. The objects present in warehouses are quite textureless. Therefore, the number of features will be less, making the feature matching methods less reliable. Also, due to the presence of multiple instances of the object, It can be challenging to select features that correspond to a single object.

The performance of CNN-based algorithms relies on extensive training on large datasets. However, the large variety of objects in the warehouse industry, its challenging to train a

CNN for all types of objects. Therefore, we aim to develop a method that requires fewer features and can easily be deployed for objects with different dimensions.

In Bazazian et al. [1], the author performs fast computation of edges by constructing covariance matrices from local neighbors and then calculating surface variation for detecting edge points, but the author has demonstrated this for edge points extraction only.

In Vohra et al. [2], the authors develop an edge and corner detection pipeline for unorganized point clouds aimed at robotic pick and place tasks. Their method first estimates surface normals at each point, then clusters points based on both normal similarity and spatial proximity to extract sharp edges and salient corners. Finally, they solve for object pose by matching these features to known CAD models. While accurate, the reliance on per-point normal computation and multiple clustering stages makes the approach computationally expensive, particularly when handling large, noisy point clouds in real time.

Our approach's main advantage is that it can be easily applied to other objects with known dimensions in less computation time.

III. Proposed Methodology

A. Edge Points Extraction

Edge extraction in unorganized 3D point clouds is critical for tasks such as object recognition and robotic grasping. Traditional approaches often rely on per-point normal estimation followed by clustering of normals to identify sharp features, which can be both sensitive to noise and computationally intensive.

Inspired by the fast and robust edge extraction framework of Bazazian et al. [1], we adopt a purely statistical method based on eigenvalue analysis of local covariance matrices, eliminating the need for explicit normal clustering and greatly simplifying the edge extraction process.

Covariance is a measure of how much each of the dimensions varies from the mean with respect to each other. For a 3 dimensional data set (X, Y, Z), the 3×3 Covariance matrix C for a sample point p(x,y,z) is given by:

$$C = \begin{bmatrix} \mathrm{Cov}(x,x) & \mathrm{Cov}(x,y) & \mathrm{Cov}(x,z) \\ \mathrm{Cov}(y,x) & \mathrm{Cov}(y,y) & \mathrm{Cov}(y,z) \\ \mathrm{Cov}(z,x) & \mathrm{Cov}(z,y) & \mathrm{Cov}(z,z) \end{bmatrix} \tag{1}$$

where, for instance Cov(x,y) is the Covariance of x, y computed as:

$$\mathrm{Cov}(x,y) = \frac{\sum_{i=1}^{k} \left(x_i - \bar{x}\right)\left(y_i - \bar{y}\right)}{n-1} \tag{2}$$

Afterwards, we explore the Eigenvalues of C: $\lambda_0 \leq \lambda_1 \leq \lambda_2$.

In [3], [4] Pauly et. al. introduce the following concept of surface variation $\sigma_k(p)$:

$$\sigma_k(p) = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{3}$$

The surface variation, $\sigma_k(p)$, for each sample point with k neighbors allows us to distinguish whether the point belongs to a flat plane or to a salient point (edge) in the point cloud as follows:

$$\sigma_k(p) = \begin{cases} 0, & \lambda_0 \approx 0 \quad \text{(flat surface)}, \\ > 0, & \text{if an edge is present.} \end{cases}$$

B. Corner Points Extraction

In this subsection, we refine our feature set by detecting salient corner points from the detected edge points in the point cloud data. We employ a 3D Harris-corner detector tailored to unorganized point clouds.

This algorithm is the 3d extension of the Harris corner detection 2d image algorithm [5] proposed by I. Laptev [6] and computes the cornerness for each pixel of the input 3d image.

As an extension of 2d case, M is defined as follows :

$$M = \sum_{x,y,z \in \mathcal{N}} \omega(x,y,z) \begin{bmatrix} I_x^2 & I_x I_y & I_x I_z \\ I_x I_y & I_y^2 & I_y I_z \\ I_x I_z & I_y I_z & I_z^2 \end{bmatrix} \tag{4}$$

With I_x, I_y, and I_z as the spatial derivatives of the extracted edge points image along the directions x, y, and z respectively, and $\omega(x,y,z)$ is a Gaussian weight in the neighbourhood $\mathcal{N}$.

The cornerness $\mathcal{C}$ is calculated at the position (u, v, w) by:

$$C(u,v,w) = \det(M) - k\left(\mathrm{trace}(M)\right)^3 \tag{5}$$

The cornerness value $C(u,v,w)$ quantifies the likelihood of a point being a corner based on the local image structure around the voxel (u,v,w). Once the cornerness values are computed for all points in the volume, the following steps are performed to extract the final set of salient corner points:

1) **Thresholding:** All points with cornerness values below a predefined threshold are discarded. This helps in removing weak corner responses caused by noise or flat regions.
2) **Non-maximum Suppression:** Among the remaining points, non-maximum suppression is applied within a local 3D neighborhood. This ensures that only the local maxima- i.e., the strongest corner responses in a given vicinity are retained.

These retained points are considered as the final detected corners. The effectiveness of the detection depends on appropriate selection of the parameters such as the Gaussian weighting function $\omega(x,y,z)$, the constant k, and the size of the neighborhood used for suppression.

Given a set of detected 3D corner points, we define

$$C = \{\, c_i = (x_i, y_i, z_i) \mid i = 1, \ldots, N \}. \tag{6}$$

We compute the coordinate-wise minima and maxima:

$$x_{\min} = \min_{1 \leq i \leq N} x_i, \quad x_{\max} = \max_{1 \leq i \leq N} x_i. \tag{7}$$

$$y_{\min} = \min_{1\le i\le N} y_i, \quad y_{\max} = \max_{1\le i\le N} y_i. \tag{8}$$

$$z_{\min} = \min_{1\le i\le N} z_i, \quad z_{\max} = \max_{1\le i\le N} z_i. \tag{9}$$

Next, form the eight vertices of the axis-aligned bounding box:

$$B = \big\{(x_a, y_b, z_c) \mid a, b, c \in \{\min, \max\}\big\}. \tag{10}$$

For each vertex $b \in B$, select the detected corner closest in Euclidean distance:

$$p^*(b) = \arg\min_{p\in C} \|p - b\|_2. \tag{11}$$

The final set of extreme corners is then

$$\{\, p^*(b) \mid b \in B\}.$$

We determine the object's eight extreme corners by first finding the smallest and largest values of the x, y, z coordinates among all detected points.

These six scalars $x_{\min}, x_{\max},\ y_{\min}, y_{\max},\ z_{\min}, z_{\max}$, define the vertices of the tightest axis-aligned bounding box around the data. Conceptually, there are eight such vertices, each corresponding to one of the two choices (minimum or maximum) along each axis. For each of these hypothetical box corners, we then search through our detected corner set and pick the single point whose Euclidean distance to that box corner is minimal. In this way, each of the eight theoretical vertices calculates the closest actual corner candidate. The result is exactly eight real points, each one the nearest detected corner to one of the box vertices, which together form a robust estimate of the object's true 3D extremities.

C. Pose Estimation

After extracting all eight extreme corners of the object, we select two orthogonal edges sharing a common vertex p_1 (the intersection) and their other endpoints p_2 and p_3. These three camera-frame points

$$p_1,\ p_2,\ p_3 \in \mathbb{R}^3$$

correspond to known local-frame corners.

$$q_1,\ q_2,\ q_3 \in \mathbb{R}^3,$$

For example

$$q_1 = \big(-\tfrac{l}{2}, -\tfrac{b}{2}, \tfrac{h}{2}\big),\ q_2 = \big(\tfrac{l}{2}, \tfrac{b}{2}, \tfrac{h}{2}\big),\ q_3 = \big(-\tfrac{l}{2}, \tfrac{b}{2}, \tfrac{h}{2}\big).$$

Inspired by Vohra et al. [2], we compute the centroids

$$\bar{p} = \frac{1}{3}\sum_{i=1}^{3} p_i, \qquad \bar{q} = \frac{1}{3}\sum_{i=1}^{3} q_i, \tag{12}$$

and assemble the cross-covariance matrix

$$H \ = \ \sum_{i=1}^{3} (p_i - \bar{p})\,(q_i - \bar{q})^T. \tag{13}$$

Performing singular value decomposition,

$$H = U\,\Sigma\,V^T, \tag{14}$$

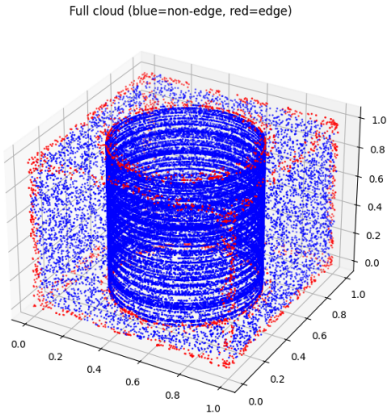

Fig. 2. Hollow cylinder inside a cube

yields the optimal rigid-body transform

$$R = V\,U^T, \qquad t = \bar{p} \ - \ R\,\bar{q}. \tag{15}$$

Here, R is the 3×3 rotation matrix aligning the object's local axes to the camera axes, and t is the translation vector from the camera origin to the object centroid.

Once the object's 6D pose (R, t) is available, the robot controller can convert this into a target pose. The robot's motion planner should interpolate a smooth path from the current arm configuration to the approach waypoint to the final grasp pose. This pose estimate provides precise positioning information that allows the robot to reliably reach, grasp, and extract the object.

IV. Experimental Results

In the initial experiment, we evaluated our method on a synthetic cube point cloud obtained from the Stanford 3D Scanning Repository, as shown in Fig. 3.

Figure 3 presents the four key stages of our cube-pose pipeline applied to the synthetic cube point cloud. First, the raw point cloud (a) shows an unstructured sampling of the cube's surface in light gray. Next, edge points are extracted using surface variation (b), highlighting the twelve edge points in red.

From these edge points, we then detect corner candidates and select the eight external points along each axis (c), marking them as green spheres against the remaining corner estimates in red. Finally, we recover the cube's pose (d) by aligning the principal axes defined by these extreme corners to the model frame.

This sequence demonstrates that our method accurately traces cube geometry and achieves precise rigid-body pose estimation.

Figure 2 demonstrates the edge-extraction performance of our method on a hollow cylinder enclosed in a cube. The complete point cloud is shown with non-edge points in blue and detected edge points in red, clearly outlining both the cube faces and the cylindrical surface. The results confirm that our approach reliably identifies edges on composite geometries, preserving sharp features on both curved and planar regions.

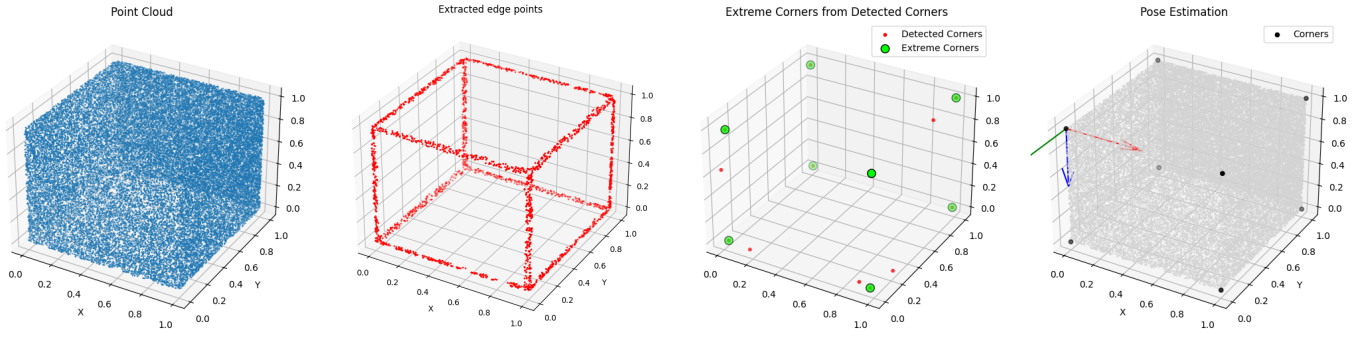

Fig. 3. Our model on Cube Point Cloud Data

A. Performance Analysis

To quantitatively compare runtime performance, we ran both our proposed pipeline and the Vohra *et al.* [2] algorithm on the same synthetic cube point cloud. Table I breaks down the computation time (in seconds) at each stage of processing.

TABLE I
COMPUTATION TIME (SEC) AT EACH STEP

Model	Edge	Corner	Pose Est.	Total Time
Our Methodology	5.520	0.573	1.850	7.943
Vohra *et al.* [2]	24.168	1.110	1.551	26.829

As shown in Table I, our pipeline reduces the most expensive edge-extraction stage from 24.17 s to 5.52 s ($\approx$ 4.4× speed-up) by using an eigenvalue-based, clustering-free approach. Corner detection time is also cut roughly in half, while pose estimation remains comparable. Overall, the end-to-end runtime falls from 26.83 s to 7.94 s ($\approx$ 3.4× faster), demonstrating that accelerating edge extraction drives almost the entire improvement without compromising downstream accuracy.

This demonstrates the computational efficiency of our approach on the same input data.

B. Performance on Different Objects

Building on the hollow cylinder in cube example shown in Figure 2, we further validated our pipeline on a variety of synthetic primitives tetrahedra, spheres, and composite assemblies by simply adjusting two key hyperparameters (the curvature-threshold τ and the Harris radius r_h) to match each object's characteristic feature scale. Without modifying the core algorithm, our eigenvalue-based edge extractor reliably isolates sharp discontinuities on tetrahedral faces, circular rims on cylinders and spheres, and linear seams in composite models. Likewise, the 3D Harris detector consistently localizes true corner points across all geometries, with non-maximum suppression tuned to the average inter-corner distance. This demonstrates that, by selecting only two intuitive parameters based on object dimensions, our framework generalizes seamlessly to new shapes, making it highly adaptable for automated pick-and-place tasks involving diverse, textureless parts.

V. CONCLUSION

In this work, we have presented a novel, fully-integrated framework for edge and corner detection in unorganized point clouds, tailored to real-time robotic pick-and-place applications. By combining a clustering-free, eigenvalue-based edge extractor with a 3D Harris corner detector and a corner-driven pose solver, our method not only accurately recovers sharp geometric features but also delivers robust 6D pose estimates. Experimental results on multiple synthetic objects (cube, hollow cylinder, tetrahedron) demonstrate that our approach traces edges and corners with high fidelity and achieves sub-voxel pose accuracy. Moreover, compared to the recent Vohra *et al.* (2021) pipeline, we observe up to a 4.4× speed-up in edge extraction and an overall 3.4× reduction in end-to-end runtime, all without sacrificing downstream accuracy. These characteristics make our framework well-suited for deployment in dynamic, cluttered environments where both precision and computational efficiency are paramount. Future work will extend the pipeline to real sensor data, nonconvex geometries, and multi-object scenes.

REFERENCES

[1] D. Bazazian, J. R. Casas, and J. Ruiz-Hidalgo, "Fast and robust edge extraction in unorganized point clouds," in *Proc. 2015 Int. Conf. Digital Image Computing: Techniques and Applications (DICTA)*, Nov. 2015.
[2] M. Vohra, R. Prakash, and L. Behera, "Edge and corner detection in unorganized point clouds for robotic pick-and-place applications," arXiv:2104.09099 [cs.RO], Apr. 2021.
[3] M. Pauly, M. Gross, and L. Kobbelt, "Efficient simplification of point-sampled surfaces," in *Proc. IEEE Visualization (VIS)*, 2002, pp. 163–170.
[4] M. Pauly, R. Keiser, and M. Gross, "Multi-scale feature extraction on point-sampled surfaces," *Comput. Graph. Forum*, vol. 22, no. 3, pp. 281–289, 2003.
[5] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. 4th Alvey Vision Conference*, 1988, pp. 147–151.
[6] I. Laptev, "On space–time interest points," *Int. J. Comput. Vision*, vol. 64, no. 2–3, pp. 107–123, 2005.
[7] S. Changali, A. Mohammad, and M. van Nieuwland, "The construction productivity imperative: How to build megaprojects better," *McKinsey Quarterly*, 2015.
[8] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.
[9] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 4490–4499.