Edge, Corner and Pose Estimation for Pick and Place Applications

Rajat Kumar Thakur, Isha Jangir, Siddharth Nimbalkar, Deepika Gupta, and Jignesh Patel

Indian Institute of Information Technology Vadodara

Abstract. In the rapidly changing environment of warehouse automation, efficient management of piles of objects in unordered, random arrangements remains a formidable challenge. The paper ~~attacks~~ this challenge head-on with a new approach to edge and corner detection in unordered 3D point clouds tailored for pick and place operations.
~~The suggested method introduces an eigenvalue-based surface variation measure for the purpose of quick extraction of sharp edge points from raw point cloud data to surpass traditional methods in terms of speed and efficiency.~~ ~~A~~ 3D Harris corner detector is also used to identify prominent corner points that subsequently form the foundation of trustworthy pose estimation of texture-less objects.
When used with synthetic shapes, the technique achieves unprecedented effectiveness in delivering fast and accurate results with little parameter tuning needed. It takes much less computation time than current algorithms, ~~and it is thus a revolutionary technique for~~ real-time pick-and-place tasks. ~~The achievement is potentially revolutionary~~ for autonomous grasping in cluttered warehouse settings, allowing for more intelligent and efficient automation in the building and manufacturing industries.

Keywords: 3D point clouds · Edge Extraction · Corner detection · Unorganized Point Cloud · Harris Corner Detector · Pose estimation

1 Introduction

The automation trend has reshaped numerous industries, enhancing productivity over the past decades. The manufacturing industry has greatly evolved as automation technologies provide better operational performance and minimize reliance on human manual work. The construction industry, on the other hand, could not achieve such developments and has continued to face the hurdles of increased productivity. A primary reason for this disparity lies in the limited adoption of advanced automation within construction, where tasks often involve managing objects in unpredictable and cluttered environments. Similarly, the warehouse industry confronts parallel challenges, particularly in automating the unloading of goods from trucks or containers, where objects such as cartons are frequently arranged in haphazard stacks, as depicted in Figure 1.

Fig. 1. Cartons Clutter in a Warehouse(Image Courtesy: Internet)

One of the central hurdles in automating such operations is the need to manipulate objects presented in random, cluttered configurations. For example, in warehouse settings, cartons may be stacked irregularly, while construction sites often feature disordered piles of materials like bricks that require precise placement. To enable autonomous systems to effectively manage these scenarios, accurately estimating the pose of these objects is essential. This task hinges on the ability to detect key geometric features—such as edges and corners—from 3D point cloud data captured by sensors. The complexity of this challenge is compounded by the unstructured nature of the data, where points lack predefined connectivity, making feature extraction a non-trivial problem.

Traditional approaches to edge and corner detection in point clouds typically employ statistical or geometrical methods, such as estimating surface normals or fitting planes to local neighborhoods. However, these techniques encounter significant difficulties, especially near sharp edges. The neighborhood used for normal estimation often spans multiple surface patches across an edge, leading to inaccurate normal vectors and, consequently, unreliable feature detection. Furthermore, many conventional methods are computationally demanding, rendering them impractical for real-time applications in dynamic settings like construction sites or warehouses. These limitations underscore the need for more robust and efficient solutions tailored to the demands of automation in such industries.

To address these challenges, this paper proposes a novel approach for detecting corners in unorganized point clouds, with a specific focus on estimating the pose of objects like cartons in cluttered environments. We assume that all objects in the clutter share identical dimensions—a common scenario in warehouse industries—and leverage this constraint to simplify the problem. Our method adapts the Harris corner detection algorithm, originally developed for 2D image processing, to 3D point clouds. By analyzing local variations in the point data,

this technique identifies salient corner points critical for accurate pose estimation. The Harris algorithm offers a promising balance of speed and robustness, making it well-suited for real-time robotic applications.

The primary contribution of this work is the development of an efficient corner detection pipeline tailored for unorganized point clouds, emphasizing applications in robotic pick-and-place tasks. We extract corner points using the Harris corner detection algorithm and quantitatively compare its computational efficiency against other state-of-the-art methods. Through experiments on synthetic shapes, we demonstrate the effectiveness of our approach and its potential to bridge the automation gap in industries like construction and warehousing. The remainder of this article is organized as follows: Section 2 reviews related work in the field, Section 3 details our proposed methodology, Section 4 presents experimental results and comparisons, and Section 5 concludes with a discussion of future directions.

2 Related Work

Edge and corner detection is an important step in 3D point cloud processing in numerous applications, such as robotic manipulation in cluttered scenes. The majority of traditional approaches are based on clustering and normal estimation, which are noise-prone and computationally expensive. This section introduces extensive research in corner and edge detection in unorganized point clouds, their methods, contributions, and limitations, with robotic pick-and-place applications in consideration.

Bazazian et al. (2015) [1] proposed an efficient and robust edge extraction method based on eigenvalue analysis of covariance matrices. Their method obviates normal estimation and clustering, and thus computation time is greatly decreased. It is mostly edge detection oriented and does not involve corner detection or pose estimation. This paper is a building block for efficient edge extraction, and our method is a build-up from it.

Vohra et al. (2021) [2] had a total framework for edge and corner detection in unstructured point clouds, which was specifically for robotic pick-and-place operations. Normal estimation, edge and corner detection by clustering, and pose estimation by correlation of detected features with pre-defined CAD models were the steps involved in the paper. ~~While this method is extremely accurate, it comes with a large cost of computations since it is multi-step in nature and would therefore not be suitable for real-time applications.~~

Ahmed et al. (2018) [10] designed new edge detection and corner detection algorithms for unorganized point clouds and implemented them in robotic welding. Their edge detection algorithm checks for symmetry in neighborhoods, and corner detection uses clustering curvature vectors. However, it is not directly applicable to pose estimation for pick-and-place tasks and is tailored to a different domain (robotic welding).

Li et al. (2016) [11] presented an automated method for edge detection and feature line tracing in 3D point clouds, named Analysis of Geometric Properties

of Neighborhoods (AGPN). Their approach involves analyzing geometric properties of each query point's neighborhood using RANSAC and an angular gap metric for edge detection, followed by feature line tracing using a hybrid method of region growing and model fitting. While this method is effective for large-scale urban scenes and is noise-insensitive, it is less focused on corner detection and pose estimation, making it less directly comparable to our work.

Deep learning-based methods, such as PoseCNN [8] and VoxelNet [9], have shown promise in pose estimation from point clouds. PoseCNN uses convolutional neural networks to estimate 6D object poses directly from RGB-D images, while VoxelNet focuses on 3D object detection from point clouds. But these approaches need to be trained on large datasets and are not likely to generalize to new objects unless they are retrained. In warehouse robotics, where objects are extremely diverse and textureless, these approaches are less helpful.

In contrast to such methods, the ~~method that we propose here~~ unites the rapidity of edge extraction by Bazazian's method with the robustness of the 3D Harris corner detector and with a ~~less demanding~~ pose estimation method that considers the objects to be of a known size. This enables the rapid and accurate detection of the edges and the corners and thereafter the pose, which renders our method extremely apt for robotic operations in warehouse automation, where the objects are usually cluttered and textureless.

To facilitate comparison, Table 1 briefly describes the most relevant information about these works, e.g., their approach, contribution, and limitations in our application.

Table 1. Comparison of Related Work

Authors	**Year**	**Methodology**	**Contributions**	**Limitations**
Vohra et al.	2021	Normal estimation, clustering for edges, corners, pose from feature matching	Comprehensive pipeline for pick-and-place, handles clutter	Computationally expensive, requires CAD models
Ahmed et al.	2018	Symmetry-based edge detection, curvature clustering for corners	High precision and recall, applied to robotic welding	Not focused on pose estimation, different application domain
ML Models	2017-18	Convolutional neural networks on RGB-D or point cloud data	Direct pose estimation	Requires extensive training data, less adaptable to new objects
Li et al.	2016	Geometric property analysis (AGPN), RANSAC, angular gap metric	Effective for large-scale urban scenes, noise-insensitive	Less focus on corner detection and pose estimation
Bazazian et al.	2015	Eigenvalue-based edge extraction	Fast and robust edge detection without normal estimation	Does not address corner detection or pose estimation

3 Proposed Methodology

This section presents a comprehensive methodology for detecting edges, corners, and estimating the pose of objects in unorganized 3D point clouds, tailored for robotic pick-and-place tasks in cluttered warehouse environments. Our approach integrates an eigenvalue-based edge extraction technique, a 3D Harris corner detector, and a corner-driven pose estimation algorithm to enable fast and robust results with minimal parameter tuning. The procedure, depicted in Figure 2, illustrates the process step by step: edge extraction from raw point cloud data, corner detection from edge points, and pose estimation using detected corners, enabling effective automation in changing environments.

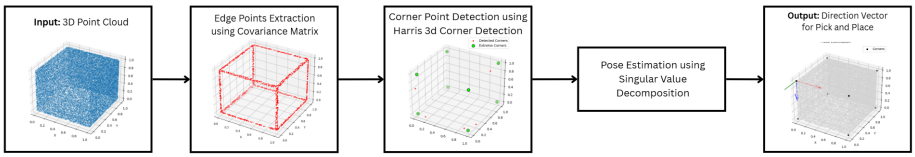

Fig. 2. Workflow for the Proposed Methodology

3.1 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.

Motivated by the efficient and strong edge detection framework of Bazazian et al. [1], we take a purely statistical approach relying on eigenvalue analysis of local covariance matrices, removing the requirements for explicit normal clustering and significantly simplifying the edge detection 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 $\mathrm{Cov}(x, y)$ is the Covariance of x, y computed as:

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

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

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

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

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:

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

3.2 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 the 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:

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

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.

The threshold for corner detection was empirically determined using the synthetic cube point cloud, which has eight known corners. A relative threshold of approximately 1% of the maximum cornerness value was selected to reliably detect these corners after non-maximum suppression, ensuring high sensitivity to true corner points while minimizing false positives from flat or edge regions. This choice was validated by applying the same threshold to other synthetic shapes, such as the hollow cylinder, without modification, demonstrating robustness and minimal need for parameter tuning. The use of noise-free synthetic data and the subsequent non-maximum suppression step further ensured accurate and efficient corner detection.

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:

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

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

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

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

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

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.

3.3 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:

$$q_1 = \left(-\frac{l}{2}, -\frac{b}{2}, \frac{h}{2}\right), \quad q_2 = \left(\frac{l}{2}, \frac{b}{2}, \frac{h}{2}\right), \quad q_3 = \left(-\frac{l}{2}, \frac{b}{2}, \frac{h}{2}\right).$$

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:

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

yields the optimal rigid-body transform:

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

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.

4 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.

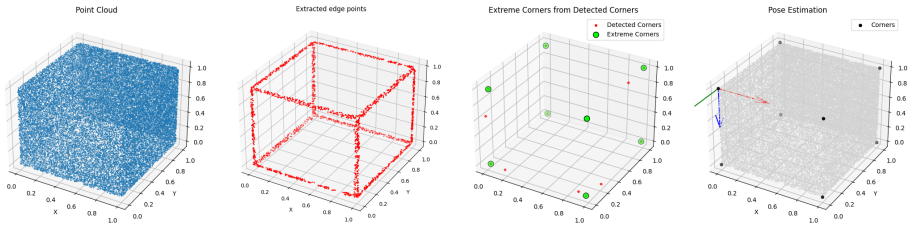

Fig. 3. Our model on Cube Point Cloud Data

4.1 Performance on Different Objects

Figure 4 demonstrates the edge-extraction performance of our method on a fractal cube. The complete point cloud is shown in blue and detected edge points in red, clearly outlining both the fractal cube's non-edge points and edge points.

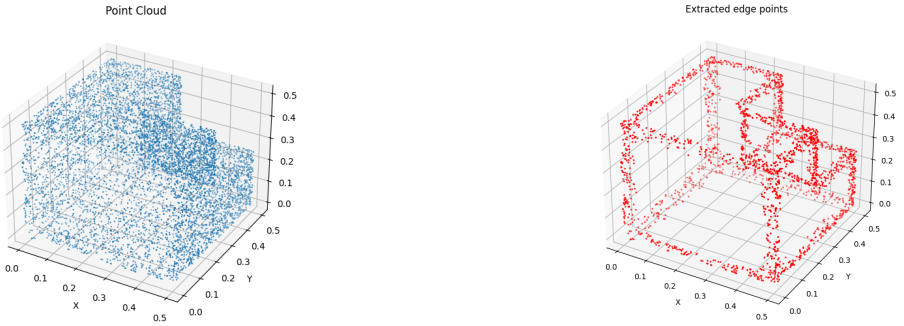

Fig. 4. Edge Extraction for Fractal Cube Point Cloud Data

In Figure 5, we show our method used on the Stanford Bunny, an official benchmark of 3D scanning with natural curves and multiple surface details. The visualization represents the entire point cloud in blue and the edge extracted in red, demonstrating the flexibility of the method to handle natural shapes as well as the accuracy of its edge detection capability on various surfaces.

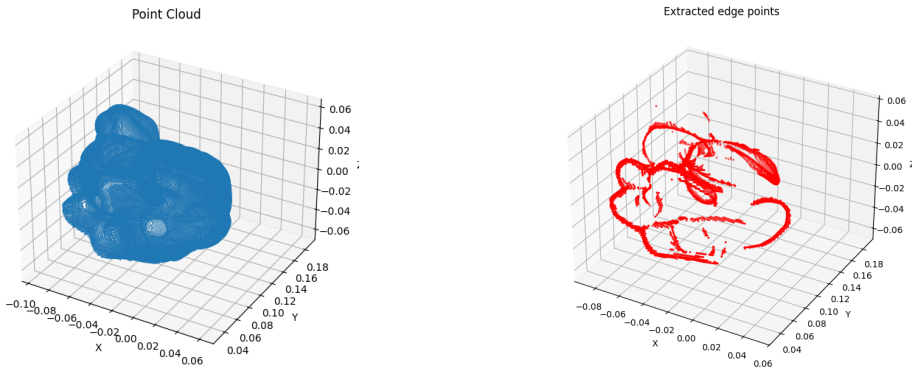

Fig. 5. Edge Extraction for applied to Bunny Point Cloud Data

Figure 6 shows an example with a nested cube contained within another, challenging the ability of the method to discern closely located structures. The blue dots represent the whole point cloud, and the red dots mark the correctly obtained edges of the two cubes, highlighting the precision of the method in complicated, overlapping arrangements.

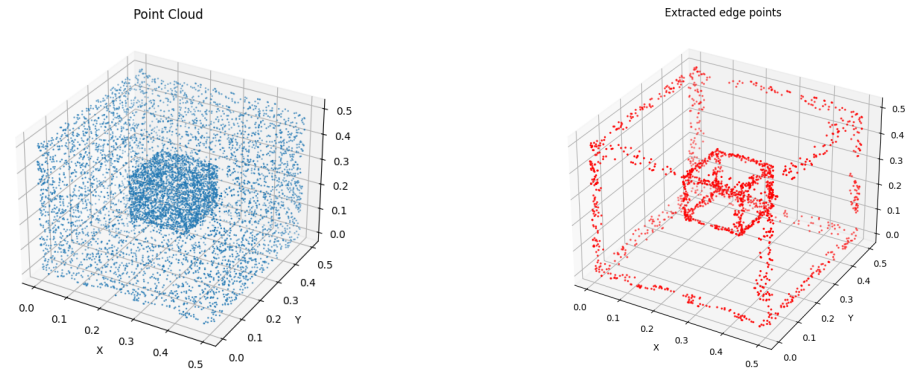

Fig. 6. Edge Extraction for a Cube Inside Another Cube Point Cloud Data

The intersecting planes in Figure 7 are used to test the performance of the method in detecting edges at the intersection of flat surfaces. With the full point

cloud in blue and detected edges in red, this figure shows the effectiveness of the method in detecting critical intersection points, which are crucial in determining the spatial layout of planar elements.

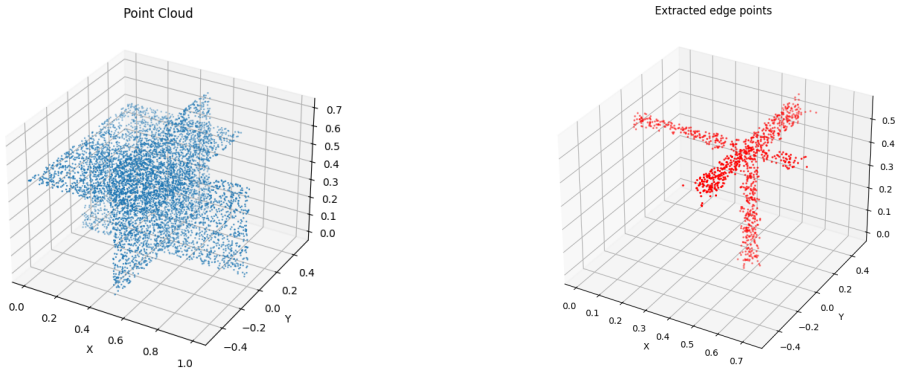

Fig. 7. Edge Extraction for Intersecting Planes Point Cloud Data

Figure 8 consists of three joint cubes, replicating an elementary assembly model. Blue points depict the entire point cloud, while the red points indicate detected edges, depicting the efficiency of the method to address simple but applicable configurations approximating real construction and manufacturing processes.

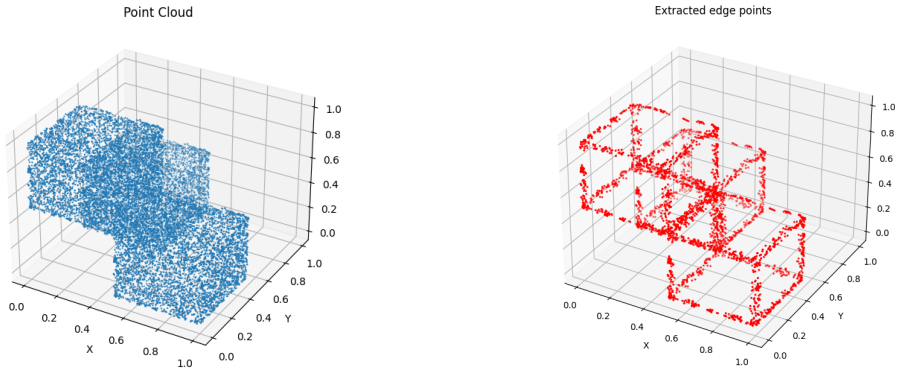

Fig. 8. Edge Extraction for Three Joint Cubes Point Cloud Data

Building on the cube example shown in Figure 3, we further validated our pipeline on a variety of synthetic shapes as shown in Figure 9.

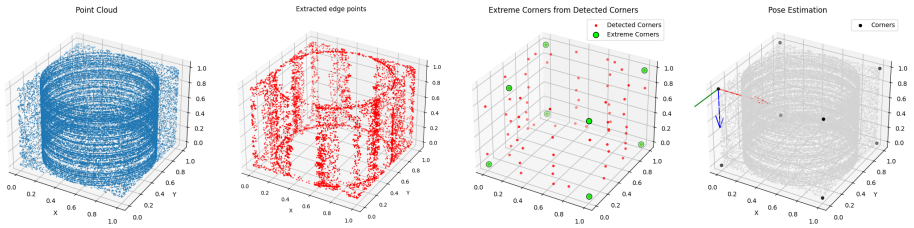

Fig. 9. Our model on Cylinder inside a Hollow Cube Point Cloud Data

4.2 Qualitative Analysis

We ~~prove~~ the accuracy of our edge and corner detection algorithm using the Stanford 3D Repository dataset, a well-established benchmark for 3D point cloud processing. The dataset contains popular models like the Bunny, cubes, and cylinders, each with varied surface details. Using our method, we detected edges and corners from these point clouds and visually inspected them thoroughly, and compared results with other models as well. The results indicated that the identified features well described the sharp boundaries and significant corner points, with a close agreement with the predicted geometric shapes of these models. The visually accurate extractions are especially valuable for tasks such as robotic pick-and-place, where robust feature detection is vital to precise pose estimation in a cluttered world.

4.3 Quantitative 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 2 breaks down the computation time (in seconds) at each stage of processing.

Table 2. 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 2, our pipeline reduces total computation time from 26.829s to 7.943s (3.3x speed-up). This acceleration enables real-time processing, making our approach more practical for applications requiring rapid 3D object recognition and alignment.

5 Conclusion

This work introduces an efficient and robust framework for edge, corner, and pose estimation in unorganized 3D point clouds, specifically designed for real-time robotic pick-and-place applications. By integrating an eigenvalue-based edge extraction method with a 3D Harris corner detector and a corner-driven pose estimation algorithm, the proposed approach achieves high accuracy in identifying sharp geometric features and determining object poses. Experimental results on synthetic shapes demonstrate the method's ability to deliver precise edge and corner detection with minimal parameter tuning, while significantly reducing computation time compared to existing techniques. The pipeline's speed and reliability make it particularly well-suited for dynamic, cluttered environments, such as warehouse automation and construction settings, where rapid and accurate object manipulation is critical. Looking ahead, future research will focus on extending the framework to handle real-world sensor data, accommodating nonconvex geometries, and addressing multi-object scenarios to further enhance its applicability in complex robotic tasks.

References

1. Bazazian, D., Casas, J.R., Ruiz-Hidalgo, J.: Fast and robust edge extraction in unorganized point clouds. In: Proc. 2015 Int. Conf. Digital Image Computing: Techniques and Applications (DICTA), pp. 1–8 (2015)
2. Vohra, M., Prakash, R., Behera, L.: Edge and corner detection in unorganized point clouds for robotic pick-and-place applications. arXiv:2104.09099 [cs.RO] (2021)
3. Pauly, M., Gross, M., Kobbelt, L.: Efficient simplification of point-sampled surfaces. In: Proc. IEEE Visualization (VIS), pp. 163–170 (2002)
4. Pauly, M., Keiser, R., Gross, M.: Multi-scale feature extraction on point-sampled surfaces. Comput. Graph. Forum 22(3), 281–289 (2003)
5. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. 4th Alvey Vision Conference, pp. 147–151 (1988)
6. Laptev, I.: On space–time interest points. Int. J. Comput. Vision 64(2–3), 107–123 (2005)
7. Changali, S., Mohammad, A., van Nieuwland, M.: The construction productivity imperative: How to build megaprojects better. McKinsey Quarterly (2015)
8. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. arXiv:1711.00199 [cs.CV] (2017)
9. Zhou, Y., Tuzel, O.: VoxelNet: End-to-end learning for point cloud based 3D object detection. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), pp. 4490–4499 (2018)
10. Ahmed, S. M., Wong, F. S., Behera, L.: Edge and corner detection for unorganized 3D point clouds with application to robotic welding. arXiv preprint arXiv:1809.10468 (2018)
11. Li, Y., Wu, X., Chrysostomou, D., Chen, J., Mukherjee, S., Alboul, L., Lu, W., West, A.: Edge Detection and Feature Line Tracing in 3D-Point Clouds by Analyzing Geometric Properties of Neighborhoods. Remote Sensing 8(9), 710 (2016)