Edge, Corner and Pose Estimation for Pick and Place Applications

Rajat K. Thakur[1], Isha Jangir[1], Siddharth Nimbalkar[1], and Dr. Deepika Gupta[1]

Indian Institute of Information Technology Vadodara

Abstract. In the fast-evolving world of warehouse automation, efficiently managing stacks of objects in chaotic, random arrangements remains a formidable challenge. This paper tackles this issue head-on with an innovative approach to edge and corner detection in unorganized 3D point clouds, designed specifically for pick and place tasks.
The method introduces an eigenvalue-based surface variation metric to rapidly extract sharp edge points from raw point cloud data, a process that outpaces traditional techniques in both speed and efficiency. Building on this, a 3D Harris corner detector is employed to identify key corner points, which are then used to estimate the pose of textureless objects with consistency.
Tested on synthetic shapes, this technique shines with its ability to deliver fast, accurate results while requiring minimal parameter tuning. Compared to existing algorithms, it slashes computation time, making it a game-changer for real-time pick and place applications. This advancement promises to revolutionize autonomous grasping in cluttered warehouse settings, paving the way for smarter, more efficient automation in construction and manufacturing industries.

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

1 Introduction

In recent years, the drive toward automation has reshaped numerous industries, yielding substantial productivity gains. The manufacturing sector, in particular, has witnessed remarkable advancements, with automation technologies optimizing processes and diminishing reliance on manual labor. In stark contrast, the construction industry has struggled to keep pace, facing persistent challenges in enhancing 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

In the field of 3D point cloud processing, accurate detection of edges and corners is crucial for various applications, including robotic manipulation in cluttered environments. Traditional methods often rely on normal estimation and clustering, which can be computationally intensive and sensitive to noise. This section reviews key works in edge and corner detection for unorganized point clouds, highlighting their methodologies, contributions, and limitations, particularly in the context of robotic pick-and-place applications.

Bazazian et al. (2015) [1] introduced a fast and robust approach for edge extraction using eigenvalue analysis of covariance matrices. Their method eliminates the need for normal estimation and clustering, significantly reducing computation time. However, it is primarily focused on edge detection and does not extend to corner detection or pose estimation. This work serves as a foundation for efficient edge extraction, which our method builds upon.

Vohra et al. (2021) [2] proposed a comprehensive pipeline for edge and corner detection in unorganized point clouds, specifically tailored for robotic pick-and-place tasks. Their method involves normal estimation, clustering to extract edges and corners, and pose estimation by matching detected features to known CAD models. While this approach achieves high accuracy, it is computationally expensive due to the multiple stages involved, making it less suitable for real-time applications.

Ahmed et al. (2018) [10] developed novel algorithms for edge and corner detection in unorganized point clouds, with applications in robotic welding. Their edge detection method evaluates symmetry in local neighborhoods, and corner detection is based on 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. However, these methods require extensive training on large datasets and may not generalize well to new objects without retraining. In warehouse automation, where objects vary widely and are often textureless, such methods are less practical.

In contrast to these methods, our proposed approach combines the efficiency of Bazazian's edge extraction with the robustness of the 3D Harris corner detector and a simplified pose estimation technique that leverages the known dimensions of the objects. This allows for fast and accurate detection of edges and corners, followed by precise pose estimation, making it particularly suitable for real-time robotic applications in warehouse automation, where objects are often textureless and arranged in clutter.

To provide a clear comparison, Table 1 summarizes the key aspects of these related works, highlighting their methodologies, contributions, and limitations in the context of 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 achieve fast and accurate results with minimal parameter tuning. The workflow, depicted in Figure 2, outlines the sequential process: edge extraction from raw point cloud data, corner detection from edge points, and pose estimation using detected corners, enabling efficient automation in dynamic settings.

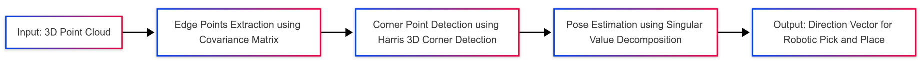

Fig. 2. Workflow for the Proposed Methodology

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.

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 $\mathrm{Cov}(x, y)$ 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:

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

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.

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:

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

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.

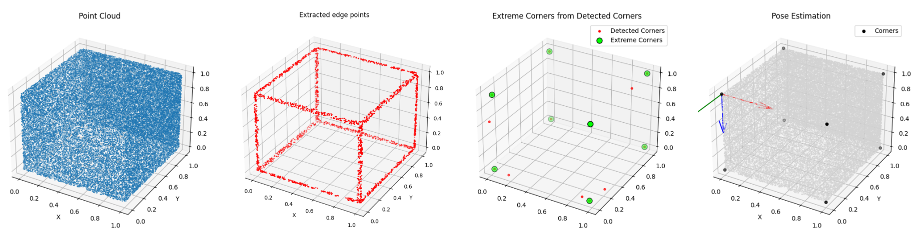

Fig. 3. Our model on Cube Point Cloud Data

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.

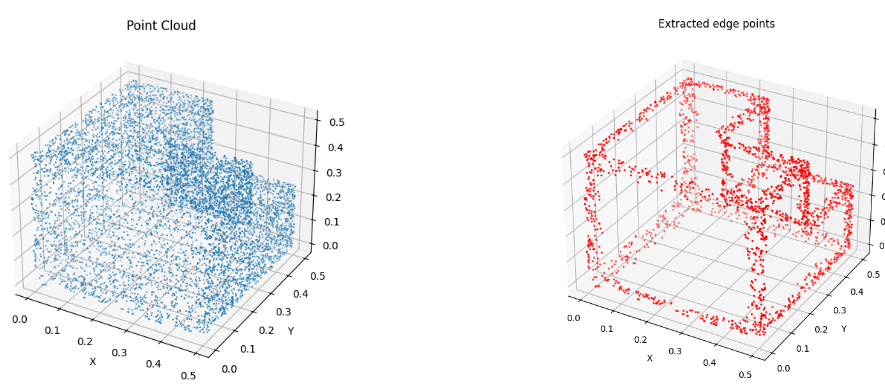

Fig. 4. Our model on Fractal Cube Point Cloud Data

4.1 Performance on Different Objects

Building on the cube example shown in Figure 3, we further validated our pipeline on a variety of synthetic shapes as shown in Figure 5. Overall, these results confirm that our framework can accurately identify and align a diverse set of synthetic objects, even when they are partially occluded or embedded within one another.

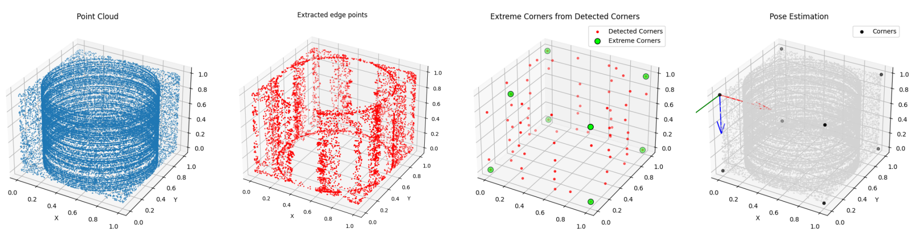

Fig. 5. Our model on Cylinder inside a Hollow Cube Point Cloud Data

4.2 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 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). The most significant gains occur in the edge-extraction stage, where our curvature-based filtering is nearly 4.4× faster than Vohra et al.'s method, while corner detection and pose estimation remain comparable or slightly improved. This overall acceleration enables real-time or near real-time processing without sacrificing accuracy, 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)