Accelerated Feature and Pose Estimation for Time-Critical Pick-and-Place Applications

Rajat Kumar Thakur[1], Isha Jangir[1], Siddharth Nimbalkar[1], Deepika Gupta[1], and Jignesh Patel[1]

Department of Computer Science and Engineering,
Indian Institute of Information Technology Vadodara-International Campus Diu, India
{202211070,202211031,22011057,deepika_gupta,jignesh_patel}@diu.iiitvadodara.ac.in

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 addresses this challenge with a novel approach to detect the geometric features such as edges and corners in the unordered 3D point clouds tailored for robotic operations.
The proposed method employs an eigenvalue-based surface variation measure to rapidly extract sharp edge points from raw point cloud data, offering improved speed and efficiency compared to traditional approaches. Additionally, 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 as compared to the previously reported algorithms. This makes it an efficient transformative tool for real-time pick-and-place tasks. This advancement helps autonomous grasping in cluttered warehouse settings, allowing for more intelligent and efficient automation in the building and manufacturing industries.

Keywords: Point Cloud · Edge Extraction · Corner detection · Pose estimation.

1 Introduction

While automation has greatly enhanced manufacturing within the industry, construction and warehousing lag behind because of the challenge in handling objects in unstructured and dynamic environments. One of the challenges is automating operations like unloading randomly pile-up containers, which requires accurate estimation of object poses. This is based on the strong extraction of geometric features, such as edges and corners, from 3D point cloud data; a challenging task given the unstructured nature of data [1]. Conventional geometric techniques employed for feature detection in point clouds tend to be unreliable close to sharp edges and computationally expensive for real-time processing in applications [3]. Deep learning methods, particularly, exhibit greater accuracy but need large annotated datasets and are prone to generalizing to unseen, texture-less objects in dense clutter. These downsides justify the demand for an efficient alternative that is robust and efficient for industrial automation.

To fill these gaps, we introduce a new corner detection algorithm for unorganized point clouds through the application of Harris corner detection to 3D data. Our method is for scenes where objects like cartons in a warehouse are of equal size. Through local point variations, the Harris algorithm effectively detects corner points essential for pose estimation and achieves a balance between speed and robustness appropriate for real-time robotics. The main contribution of this paper is an effective corner detection pipeline whose performance and computational efficiency are demonstrated through experiments on synthetic shapes and compared against state of the art.

2 Related Work

A number of geometric solutions have been proposed. Bazazian et al. (2015) [2] proposed a computationally efficient edge detection method using eigenvalue analysis that dispenses with the estimation of normals but not corner detection or pose estimation. Conversely, Vohra et al. (2021) [3] proposed a full pipeline for edge/corner detection and pose estimation via feature matching with CAD models but its multi-stage process is too slow for real-time applications. Other work, such as that of Ahmed et al. (2018) [11] in robotic welding and Li et al. (2016) [12] in large cityscapes, are application-oriented and are less concerned with corner-based pose estimation for pick-and-place applications.

Deep learning models such as PoseCNN [9] and VoxelNet [10] proved to be strong in pose estimation but are severely flawed. They require enormous training on large-scale datasets and fail to generalize to unseen or texture-deficient objects, which are common in warehouses.

The technique we suggest overcomes these limitations by combining the speed of eigenvalue-based methods with the stability of the 3D Harris corner detection algorithm. Given the presence of objects of some minimum width, our technique achieves corner detection at high speed and accuracy for pose estimation tasks and is hence highly well-suited for robotic application in unstructured, texture-poor warehouse environments.

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

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

enable fast and robust results with minimal parameter tuning. The procedure, depicted in Figure 1, 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.

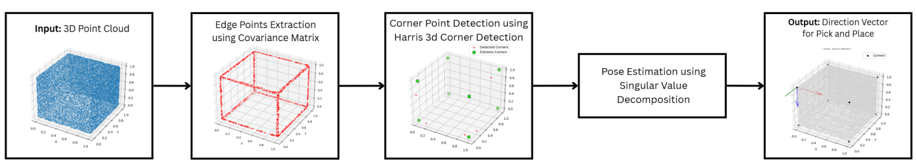

Fig. 1: Workflow for the Proposed Methodology

3.1 Edge Points Extraction

Motivated by the efficient and strong edge detection framework of Bazazian et al. [2], 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 quantifies how pairs of dimensions jointly deviate from their means. For a three-dimensional dataset (X, Y, Z), the 3×3 covariance matrix C for a sample point $p = (x, y, z)$ is given by:

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

Here, for example, $\mathrm{Cov}(X, Y)$ denotes the covariance between X and Y, calculated as:

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

We then explore the Eigenvalues of C: $\lambda_0 \leq \lambda_1 \leq \lambda_2$.

In Pauly et al. [4, 5], the 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 $\sigma_k(p)$, computed over the k nearest neighbors of point, indicates whether it lies on a planar region or corresponds to a salient feature (edge), 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

The 3D Harris corner detection algorithm is the 3D extension of the Harris corner detection 2D image algorithm [6] proposed by I. Laptev [7] 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:

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

The cornerness value $C(u, v, w)$ measures the likelihood of a voxel (u, v, w) being a corner based on its local image structure. To extract salient corners, we apply: (1) thresholding to discard low cornerness values, and (2) Non-maximum suppression to retain only local maxima within a 3D neighborhood. A relative threshold of approximately 1% of the maximum cornerness value was selected to reliably detect these corners flat or edge regions. This choice was validated by applying the same threshold to other synthetic shapes 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 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:

$$p_1, p_2, p_3 \in \mathbb{R}^3 \quad , \quad q_1, q_2, q_3 \in \mathbb{R}^3$$

Inspired by Vohra et al. [3], 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.

4 Experimental Results

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

Figure 2 illustrates the four stages of our pipeline on a synthetic cube point cloud. (a) shows the raw, unstructured input. (b) highlights edge points in red using surface variation. (c) identifies corner candidates, with selected extreme corners marked in green. (d) aligns these corners to recover the cube's pose relative to the model frame.

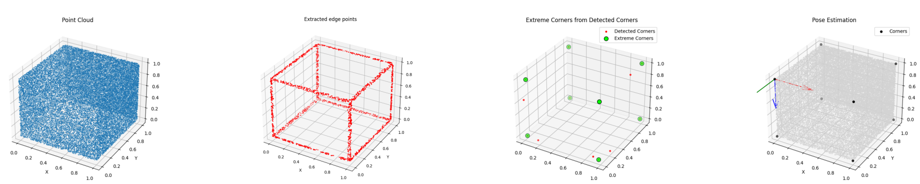

Fig. 2: Our model on Cube Point Cloud Data

Building on the cube example shown in Figure 2, we further validated our pipeline on a variety of synthetic shapes as shown in Figure 3.

In figure 4 we demonstrate the accuracy of our edge and corner detection algorithm using the Stanford 3D Repository dataset. 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

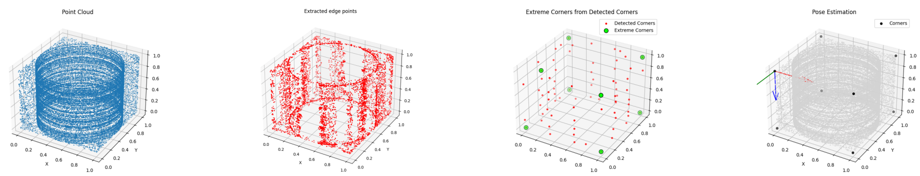

Fig. 3: Our model on Cylinder inside a Hollow Cube Point Cloud Data

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.

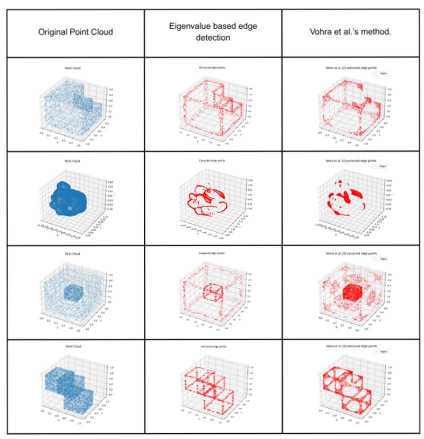

Fig. 4: Edge extraction methods on Stanford 3D Repository Cloud Data

To quantitatively compare runtime performance, we ran both our proposed pipeline and the Vohra et 1al. [3] algorithm on the same synthetic cube point cloud. Table 2 breaks down the computation time (in seconds) at each stage of processing. As shown, 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 real time applications.

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

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. [3]	24.168	1.110	1.551	26.829

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 where rapid and accurate object manipulation is critical. 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. Changali, S., Azam M., Mark van N.: The construction productivity imperative. In: McKinsey Quarterly, pp. 1–10 June (2015).
2. 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 (2015)
3. 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)
4. Pauly, M., Gross, M., Kobbelt, L.: Efficient simplification of point-sampled surfaces. In: Proc. IEEE Visualization (VIS), pp. 163–170 (2002)
5. Pauly, M., Keiser, R., Gross, M.: Multi-scale feature extraction on point-sampled surfaces. Comput. Graph. Forum 22(3), 281–289 (2003)
6. Harris, C., Stephens, M.: A combined corner and edge detector. In: Proc. 4th Alvey Vision Conference, pp. 147–151 (1988)
7. Laptev, I.: On space–time interest points. Int. J. Comput. Vision (2005)
8. Changali, S., Mohammad, A., van Nieuwland, M.: The construction productivity imperative: How to build megaprojects better. McKinsey Quarterly (2015)
9. Xiang, Y., Schmidt, T., Narayanan, V., Fox, D.: PoseCNN: A convolutional neural network for 6D object pose estimation in cluttered scenes. arXiv:1711.00199 (2017)
10. 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 (2018)
11. Ahmed, S. M., Wong, F. S., Behera, L.: Edge and corner detection for unorganized 3D point clouds with application to robotic welding.(2018)
12. Li, Y., Wu, X. :Edge Detection and Feature Line Tracing in 3D-Point Clouds by Analyzing Geometric Properties of Neighborhoods. Remote Sensing (2016)
13. Liu, J., Sun, W., Yang, H. : Deep Learning-Based Object Pose Estimation: A Comprehensive Survey. arXiv:2405.07801 [cs.CV], 2024.