Segmentation and Localization of Randomly Arranged Objects Using ICP for Tabletop Environments

Yicheng Jiang
Robotics
University of Michigan - Ann Arbor
Ann Arbor, United States
valeska@umich.edu

Zijie Chen
Robotics
University of Michigan - Ann Arbor
Ann Arbor, United States
chenzj@umich.edu

***Abstract*—This project focuses on segmenting and localizing randomly arranged objects on a tabletop using point cloud processing techniques. The proposed method begins by filtering out the tabletop using a RANSAC-based plane model to isolate the objects from the background. Following this, DBSCAN clustering is applied to segment the remaining point clouds into individual objects. For alignment of point clouds, Point Feature Histogram (PFH) descriptors are employed in conjunction with RANSAC to provide an initial transformation. To refine the alignment, three variants of the Iterative Closest Point (ICP) algorithm, point-to-point, point-to-plane, and generalized ICP, are evaluated for their effectiveness. The approach successfully identifies and localizes a random subset of objects placed in the scene, demonstrating its robustness in handling arbitrary configurations of objects with known point cloud models on the tabletop. This work provides a foundation for applications that require object recognition and spatial awareness in dynamic environments, such as robotic manipulation and automation systems.**

I. Introduction

Understanding and interacting with unstructured environments is a fundamental challenge in robotics, particularly for tasks requiring object manipulation. Effective segmentation and localization of objects are critical for enabling robots to perceive their surroundings accurately and perform actions such as grasping and pushing. These capabilities are especially important in scenarios where objects are randomly placed, such as in dynamic and cluttered environments like warehouses or home settings.

In this project, we study the problem of segmenting and localizing randomly arranged objects on a tabletop. Given a set of known objects and their point cloud models, a random subset is placed on the table in arbitrary configurations. The task is to identify the objects present in the scene and determine their positions and orientations. The setup assumes access to point cloud data obtained from a 3D sensor. The primary challenge lies in robustly filtering out the table surface, segmenting the objects, and accurately aligning the segmented clusters with the known object models.

To address these challenges, we employ a combination of point cloud processing techniques, including RANSAC-based plane filtering, DBSCAN clustering for segmentation, and alignment methods leveraging PFH descriptors and ICP variants. The goal is to develop a robust framework for robotic perception and manipulation tasks by achieving accurate segmentation and localization of objects in diverse arrangements and configurations.

II. Literature Review

This section briefly reviews foundational work in DBSCAN clustering, Persistent and Fast Point Feature Histograms (PFH/FPFH), and variations of the Iterative Closest Point (ICP) algorithm, including Point-to-Point ICP, Point-to-Plane ICP, and Generalized ICP. These foundational methods collectively inform the project's approach to segmenting and localizing objects in randomly arranged tabletop scenes.

A. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm specifically developed for spatial data and is capable of identifying an unknown number of clusters a priori [1]. It defines clusters based on density connectivity, where a point belongs to a cluster if it is density-reachable from any other point in the same cluster. A point is considered density-reachable if it lies within the epsilon radius (ϵ) of a core point, which itself must have at least a minimum number of neighboring points ($MinPts$). DBSCAN effectively discovers clusters of arbitrary shapes and handles noise. Its ability to avoid reliance on convex cluster boundaries further enhances its ability to efficiently capture non-linear and irregular patterns in complex datasets.

B. PFH and FPFH

Persistent Point Feature Histograms (PFH) are robust descriptors that capture the local geometry around a point by encoding angular and spatial relationships between neighboring points into histograms [2]. The PFH computation involves estimating surface normals for the k-nearest neighbors of a query point and computing geometric features, such as angles and distances, between all possible pairs of points within this neighborhood. These features are then binned into histograms

to represent the local geometry of the point. PFH is invariant to transformations and resilient to noise, which makes it highly effective for point cloud alignment tasks.

Despite its robust power, the computational complexity of PFH, which is $O(n \cdot k^2)$, limits its use for large-scale datasets and real-time applications. To address these limitations, Fast Point Feature Histograms (FPFH) were introduced as a computationally efficient alternative [3]. FPFH reduces the complexity to $O(n \cdot k)$ by simplifying the histogram generation process. Instead of considering all pairwise relationships in the neighborhood, FPFH computes a Simplified PFH (SPFH) for each point and then aggregates these histograms with those of its neighbors using weighted contributions. While FPFH sacrifices some geometric detail, it retains sufficient geometric representation capabilities for applications like initial alignment in point cloud registration, where efficiency is critical.

C. ICP variants

The Iterative Closest Point (ICP) algorithm aligns point clouds by iteratively minimizing the distance between corresponding points. The standard Point-to-Point ICP minimizes the Euclidean distance between paired points, which provides a simple yet effective approach for rigid body alignment [4]. While computationally efficient, it assumes perfectly overlapping point clouds and struggles with local minima or noise. Point-to-Plane ICP addresses these issues by minimizing the error along surface normals [4].

Generalized ICP extends traditional ICP methods by adopting a probabilistic framework that leverages local geometric structures from both point clouds [4]. By incorporating covariance matrices to model these structures, it accounts for noise and uncertainties more effectively. This enhances robustness against incorrect correspondences and improves alignment accuracy compared to Point-to-Point and Point-to-Plane ICP. It is particularly effective in handling partially overlapping point clouds and complex surface geometries.

III. Methodology

A. Problem setup

This study focuses on segmenting and localizing objects placed randomly on a tabletop. The input consists of 3D point clouds, which includes both the tabletop surface and objects. The problem is formulated under the assumptions that:

1) The tabletop is flat, uniform, and parallel to the XY-plane of the 3D coordinate system.
2) The tabletop point cloud constitutes the majority of points in the scene, represents the largest planar structure and dominates the overall point distribution.
3) Objects are non-overlapping and fully visible.
4) The point cloud resolution is sufficient for distinguishing individual objects.
5) The objects are rigid and non-deformable. This assumption simplifies the alignment process by ensuring that the object shapes do not change, and their geometric features remain consistent with those of the pre-defined models.

B. Data processing

In order to have a cleaner scene for object segmentation, The tabletop surface is detected and removed from the input point cloud using a RANSAC-based plane segmentation approach. RANSAC identifies the surface with the maximum number of inlier points by iteratively fitting plane models to randomly sampled points and minimizing residual errors. Leveraging Assumption 2, which guarantees that the largest planar surface corresponds to the tabletop, the algorithm reliably estimates a plane model representing the tabletop. Points classified as inliers to this planar surface are subsequently removed from the point cloud. The remaining points, considered outliers relative to the plane, are retained as object point clouds for further processing.

Based on Assumption 1 outlined in the problem setup, the tabletop is considered parallel to the XY-plane. However, due to the presence of object point clouds above the tabletop surface, the estimated plane model may deviate slightly from this ideal, which results in small non-zero components in the x and y directions of the plane normal vector. To mitigate the influence of these outliers, the x and y components of the estimated normal are adjusted back to zero to align the plane with the assumed ideal orientation. The inlier threshold parameter is then carefully tuned to ensure all points belonging to the tabletop surface are effectively excluded from the remaining object-only scene. This adjustment ensures robust segmentation while minimizing the impact of object points on the plane model estimation.

C. Object segmentation

After preprocessing, the filtered point cloud undergoes segmentation to divide it into individual clusters, each representing an object. DBSCAN is selected for its robustness in handling irregular shapes, varying densities, and an unknown number of clusters. The segmentation process involves tuning two critical parameters: the neighborhood radius (ϵ) and the minimum number of points ($MinPts$) required to define a core point.

In this project, ϵ is adjusted carefully to match the scale of the objects in the scene, which ensures meaningful clusters are identified without over-segmenting. The value of $MinPts$ is chosen based on empirical observations to achieve a balance between sensitivity to sparse clusters and exclusion of noise points. Manual parameter tuning optimizes the segmentation process for the specific characteristics of the dataset selected in this project.

Another key feature of DBSCAN is its ability to separate objects that are closely spaced. It uses density-based grouping to distinguish adjacent objects. Points that do not meet the density criteria are classified as noise. This characteristic is particularly beneficial when multiple objects are randomly placed on a tabletop, with some positioned in close proximity to one another. Such functionality is essential to prevent the clustering of adjacent objects into a single group, which would otherwise increase the likelihood of errors in subsequent alignment tasks.

By employing DBSCAN for segmentation, the algorithm effectively generates distinct and meaningful clusters, each representing a single object. This step is critical for isolating objects from the overall scene, which ensures accurate and efficient processing in subsequent stages.

D. Initial alignment

For a single target object in the scene, initial alignment with any model is achieved using a RANSAC-based correspondence framework. Fast Point Feature Histogram (FPFH) descriptors serve as the primary tool to characterize local geometric features of the target object and the model. These descriptors encode the spatial relationships between points within a local neighborhood and offer a robust representation.

The integration of RANSAC with FPFH descriptors ensures a reliable method to identify correspondences between the target object and the model. FPFH descriptors are first computed for the point clouds of both the target and the model. Candidate correspondences are established by comparing descriptors based on the similarity of their histograms. However, Noise, occlusion, and other sources of error can lead to incorrect matches. Consequently, not all correspondences are valid.

To address this challenge, RANSAC evaluates the candidate matches by sampling small subsets of correspondences. For each subset, a transformation is estimated using the standard singular value decomposition (SVD) method, and its quality is assessed based on the inlier ratio. The inlier ratio quantifies the proportion of correct correspondences identified during the initial FPFH matching. A correspondence is considered correct if the transformed source point lies within a specified distance of its corresponding target point. This metric evaluates the quality of the transformation by counting the number of inliers associated with the randomly selected points. A higher inlier count indicates a more reliable transformation and effectively eliminates potential false correspondences.

The resulting transformation provides a coarse alignment between the model and the object. The combination of FPFH's descriptive capability and RANSAC's robustness ensures a reliable starting point for subsequent refinement steps in the registration pipeline.

E. Point cloud registration

To refine the initial transformations obtained during object localization, Iterative Closest Point (ICP) variants are applied. These methods align the segmented clusters with the corresponding object models by iteratively minimizing the alignment error. This study implements three ICP variants to evaluate their performance in terms of alignment accuracy and robustness:

- The baseline method, Point-to-Point ICP, minimizes the sum of squared Euclidean distances between corresponding points in the target and the source. The correspondence for each point in one point cloud is determined as the nearest neighbor in the other. The iterative process involves alternating between computing correspondences and estimating a rigid transformation using SVD.
- Point-to-Plane ICP incorporates surface normals to improve alignment precision. Instead of minimizing Euclidean distances, it minimizes the projection of alignment errors onto the surface normals of the target point cloud. Surface normals are computed during preprocessing. The transformation is estimated by solving a linear system that minimizes errors along these normals. This method is theoretically effective for objects with flat or smooth surfaces. However, accurate surface normal estimation is critical, as errors in normal estimation can result in suboptimal alignments.
- Generalized ICP (GICP) models the local structure of both point clouds using covariance matrices. This framework incorporates uncertainties in point positions and local surface properties, which are not addressed by either Point-to-Point or Point-to-Plane ICP. Based on the covariance matrices, GICP seamlessly adapts its behavior. When the covariance matrices indicate uniform uncertainty in all directions, GICP behaves like Point-to-Point ICP. Conversely, when the covariance matrices emphasize uncertainty along the plane normal direction, GICP behaves like Point-to-Plane ICP. GICP generalizes the alignment process by integrating point-wise distances and surface normal information within a probabilistic framework. This approach enables GICP to combine the simplicity and broad applicability of Point-to-Point ICP with the local geometric detail and robustness of Point-to-Plane ICP. Consequently, GICP typically matches or outperforms these two variants in most scenarios.

Each ICP variant has distinct advantages and limitations. However, all are expected to enhance the refinement of point cloud alignment following the initial alignment step.

F. Object-to-model matching

To identify the correct model for each segmented object in the scene, the point cloud alignment pipeline is applied to all known models for each object. The model that best fits each object is selected based on the root mean squared error (RMSE) of its inliers. Similar to the inliers used in the RANSAC algorithm during the initial alignment, the inliers here are determined by the distance between corresponding points to minimize the influence of outliers on the RMSE metric. Additionally, the scene may contain redundant objects that are not of interest and must be ignored. A threshold for the RMSE is empirically determined to identify and exclude such irrelevant objects. The transformation matrices of the final selected models are then used to determine the objects' locations and orientations.

IV. Experiments

A. Settings

Experiments were conducted on an Alienware m15 R3, with Intel Core i7-10875H CPU, 16 GB memory, and NVIDIA GeForce RTX 2080 Super with Max-Q Design, using the 64-bit Ubuntu 20.04.6 LTS operating system. The software

environment used is Python 3.11.10, numpy 2.1.3, matplotlib 3.9.3, scipy 1.14.1, open3d 0.18.0.

B. Dataset

The real point clouds were captured using RGB-D cameras as part of Assignment 5 in the course *ROB 498: Introduction to Manipulation*, instructed by Prof. Nima Fazeli. The dataset consists of four sets of point clouds captured from different angles, including an M-block and a tabletop. These four sets were merged into a single point cloud to reconstruct the scene, which can be visualized using Open3D's visualization tool. Additional point cloud models are provided in `.ply` format and are loaded using Open3D. The reconstructed models of the Stanford bunny and the armadillo were obtained from the Stanford 3D Scanning Repository [5]. The 3D point cloud data for the wooden chair and the golden tiger were retrieved from Artec 3D [6], [7].

The imported `.ply` data were initially downsampled and scaled to align with the dimensions of the tabletop. These point clouds were then randomly and appropriately transformed to be included in the scene. To improve realism, the point clouds were further downsampled, and Gaussian noise was applied to individual points. The complete dataset includes models of an M-block, a Stanford bunny, an armadillo, a chair, and a tiger. The original real point clouds also contain partial views of other objects. To minimize the influence of these irrelevant elements, the scene is constrained to a square region within $[-0.45, -0.45] \times [0.45, 0.45]$.

C. Procedure

For each scene, two or more objects are selected from the available models and randomly positioned on the tabletop. As described in Section III, RANSAC-based plane segmentation is applied initially to remove points corresponding to the tabletop. Subsequently, DBSCAN is used to segment the remaining points based on their density distribution. For each segmented point cloud, every model attempts alignment by performing an initial alignment followed by refinement using an ICP variant. The best fit is identified as the model yielding the lowest inlier RMSE transformation. After processing all segmented point clouds, any fit with an inlier RMSE exceeding a predefined threshold is discarded. The remaining fits are deemed correct matches, with the corresponding point clouds correctly identified as one of the objects in the models. The transformations of these models are returned. Using these transformation matrices, the locations and orientations of the objects in the scene are determined.

Key parameters for the segmentation and alignment steps were manually adjusted based on qualitative observations to ensure optimal performance. The distance threshold for plane segmentation was set to filter out the planar tabletop surface effectively and minimize the removal of points from nearby objects. In DBSCAN, neighborhood radius (ϵ) was chosen to reflect the average distance between points within an object cluster. Minimum number of points per cluster ($MinPts$) was tuned to suppress noise and small irrelevant clusters.

(a) Raw scene

(b) Filtered scene

(c) Segmented scene

Fig. 1: (a) Scene showing two randomly selected point clouds without noise, placed arbitrarily on a table alongside the M-block; (b) filtered scene with the table removed, showing only the objects placed on the table; and (c) Segmented scene showing a yellow bunny, a purple armadillo, a blue M-block, and a green bar.

The distance threshold used to count inliers was calibrated to achieve a balance between runtime and accuracy.

D. Results without noise

The objective of this experiment is to visualize and evaluate segmentation and localization performance in a noise-free environment, establishing a baseline for comparison with noisy conditions. GICP is initially employed to visualize the experimental results. Subsequently, the performances of the three ICP variants are compared.

Fig. 1 presents a sequence of steps involved in processing the tabletop scene. Figure 1a illustrates the unprocessed tabletop scene. Two randomly placed objects are shown in red. In this ideal scenario, the object surfaces are assumed to be smooth. Fig. 1b presents the scene after the table has been filtered out, leaving only the objects originally placed on the table visible. All points associated with the table are removed, while nearly all points corresponding to the objects are preserved. As shown in Fig. 1c, DBSCAN correctly assigns the object points to the appropriate cluster. Furthermore, it successfully segments the bar from the M-block, despite the close proximity.

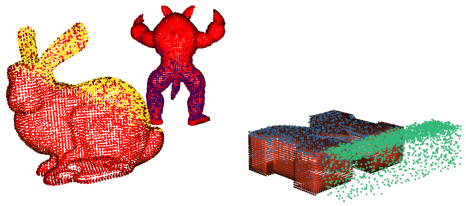


Fig. 2: Accurately aligned red models overlaid on the objects in the scene.

Fig. 2 illustrates the final alignment achieved after applying the RANSAC-based initial transformation using FPFH and subsequently performing Generalized ICP. The source models and target objects overlap, demonstrating that the alignments are accurate. No model aligns with the green bar, indicating that the inlier RMSE threshold is appropriately defined and effectively applied.

Table I presents the performance comparison of three ICP variants, with initial alignment based on FPFH serving as the baseline. The evaluation is performed on a fixed scene comprising an M-block with a bar near it, a bunny, and an armadillo. The methods are evaluated based on the total inlier RMSE associated with their best-fit models. The baseline method produces the highest total inlier RMSE. Both Point-to-Point ICP and Generalized ICP achieve the lowest total inlier RMSE, differing by only 0.0003. This result demonstrates that under ideal conditions without noise, Point-to-Point ICP is as effective as Generalized ICP. In contrast, Point-to-Plane ICP exhibits a slightly higher total inlier RMSE than the other two variants, primarily because the bunny and the armadillo feature complex surfaces with irregular geometries and sharp edges, whereas this variant performs better on planar or smooth surfaces.

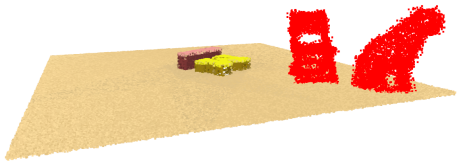
TABLE I: Comparison of Methods by Total Inlier RMSE in the Absence of Noise

Methods	Total Inlier RMSE
Initial alignment with FPFH	0.01864
FPFH + Point-to-Point ICP	0.01540
FPFH + Point-to-Plane ICP	0.01678
FPFH + Generalized ICP	0.01537

E. Results with noise

This experiment is conducted under conditions similar to those without noise, except that Gaussian noise with a variance of 0.005 is applied to each manually added object point. Initially, GICP is used to visualize the experimental results. The performance of the three ICP variants is then compared.

Fig. 3a depicts the unprocessed tabletop scene. Two randomly placed objects are highlighted in red. In this context, it



(a) Raw scene

(b) Filtered scene (c) Segmented scene

Fig. 3: (a) Scene illustrating two randomly selected point clouds with noise, arbitrarily positioned on a table alongside the M-block; (b) filtered scene with the table removed, showing only the objects initially placed on the table; and (c) segmented scene highlighting the M-block, a bar, a noisy chair, and a noisy tiger, with each cluster represented in distinct colors.

is evident that the red point clouds exhibit significant noise. In Fig. 3c, DBSCAN effectively assigns the object points to their respective clusters and successfully segments the bar from the M-block, despite their close proximity. The outlier points caused by noise are shown in black, indicating that these points are classified as noise and excluded from the object clusters. The segmentation process using DBSCAN inherently filters noise, which significantly enhances the performance of point cloud alignment in subsequent steps.

Fig. 4 illustrates the final alignment obtained after applying the RANSAC-based initial transformation with FPFH, followed by Generalized ICP. The accurate alignment of models over the noisy point clouds demonstrates the effectiveness of the approach.

Table II presents a performance comparison of three ICP variants, using initial alignment based on FPFH as the base-

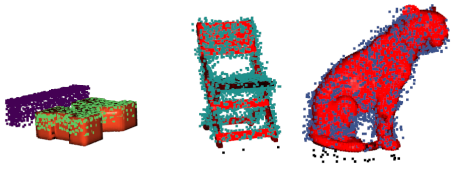


Fig. 4: Aligned red models overlaid on the objects in the scene.

line. The evaluation is conducted on a fixed scene comprising an M-block with an adjacent bar, a noisy chair, and a noisy tiger. The methods are assessed based on the total inlier RMSE of their best-fit models. Notably, the largest total inlier RMSE is produced by the Point-to-Plane ICP variant. Introducing noise to the data points appears to diminish this variant's effectiveness further. In contrast, Generalized ICP demonstrates superior performance, highlighting its robustness in handling noisy scenes.

TABLE II: Comparison of Methods by Total Inlier RMSE in the Presence of Noise

Methods	Total Inlier RMSE
Initial alignment with FPFH	0.01857
FPFH + Point-to-Point ICP	0.01747
FPFH + Point-to-Plane ICP	0.02168
FPFH + Generalized ICP	0.01458

V. Discussion

A. Strengths of the approach

This approach successfully segmented and localized objects in random arrangements. DBSCAN proved effective for identifying clusters of arbitrary shapes, while Generalized ICP delivered precise alignments even in the presence of noise. The combination of RANSAC for preprocessing and FPFH descriptors for initial alignment reduced computational overhead, enabling efficient processing of point clouds while maintaining accuracy.

B. Limitations

Although this approach demonstrates success in the experiments conducted for this project, certain limitations have been identified. First, the accuracy of matching models to objects in the scene heavily depends on their relative sizes. For instance, when the tiger model is scaled down to a size comparable to the M-block, the M-block can be mistakenly identified as a tiger in some scenarios. Furthermore, due to the limitations of the available real-world datasets, some objects were manually introduced into the scene for this project. Although techniques such as adding noise and downsampling were applied to simulate real point cloud scans, these simulated data points differ significantly from actual data. In real-world scenarios, occlusion may occur, and the scanned point clouds could exhibit a significantly different distribution from the known point cloud models. Consequently, without access to more real-world datasets, it is challenging to determine whether this approach is robust enough for practical applications.

C. Potential improvements

To address the identified limitations, several enhancements can be considered. First, incorporating scale-invariant feature extraction techniques may reduce errors in distinguishing objects of similar sizes, such as the tiger model and the M-block. Furthermore, the methodology can be improved by replacing well-reconstructed point cloud models with real-world datasets. This substitution ensures a point distribution more representative of real-world point clouds obtained from RGB-D cameras or LiDAR sensors. Robustness could be further enhanced through data augmentation techniques using synthetic occlusions or noise patterns. Lastly, advanced alignment methods, including machine learning-based object recognition or hybrid approaches that combine feature descriptors with semantic information, could improve accuracy and reliability in complex scenes. These ideas are intended to close the gap between experimental results and real-world applications.

VI. Conclusion

This project developed a robust approach for segmenting and localizing objects in a tabletop scene using point cloud data. The methodology demonstrated success in achieving accurate segmentation and alignment, though limitations in real-world applicability were identified. Paving the way for practical applications in robotics and automation, future work could address these challenges through enhanced feature extraction, real-world dataset integration, and improved alignment techniques.

Acknowledgment

The authors would like to express their sincere gratitude to Professor Nima Fazeli and GSI Andrea Sipos for their invaluable guidance, constructive feedback, and continuous support throughout this work. Their expertise in point cloud segmentation and processing has been instrumental in shaping the ideas and methodologies presented in this paper.

References

[1] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96), AAAI Press, 1996.

[2] R. B. Rusu, Z. C. Marton, N. Blodow, and M. Beetz, "Persistent Point Feature Histograms for 3D Point Clouds," in Intelligent Autonomous Systems 10, pp. 119–128, 2008.

[3] R. B. Rusu, N. Blodow, and M. Beetz, "Fast point feature histograms (fpfh) for 3d registration," 2009 IEEE International Conference on Robotics and Automation, pp. 3212–3217, 2009.

[4] A. V. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in Robotics: Science and Systems V, The MIT Press, 2010, ISBN: 9780262289801.

[5] "The Stanford 3D Scanning Repository," [Online]. Available: http://graphics.stanford.edu/data/3Dscanrep/. [Accessed: Dec. 12, 2024].

[6] "Wooden Chair HD," Artec 3D. [Online]. Available: https://www.artec3d.com/3d-models/wooden-chair-hd. [Accessed: Dec. 12, 2024].

[7] "Golden Tiger," Artec 3D. [Online]. Available: https://www.artec3d.com/3d-models/golden-tiger. [Accessed: Dec. 12, 2024].