Mapping of Primitives in Mobile Robotics Using Computer Vision

Guilherme Vaz, Ricardo Costa, Pedro Campião

FCUP - Faculty of Sciences, University of Porto
up202106775@up.pt, up20212021075547@up.pt, up202106968@up.pt

Abstract. The study presents an innovative approach to detecting and mapping geometric primitives using LiDAR data in mobile robotics. A mobile robot equipped with a LiDAR sensor is utilized to identify and map 2D shapes such as circles, rectangles, triangles, and pentagons. The method integrates the K-Means clustering algorithm for point segmentation and the Random Sample Consensus (RANSAC) algorithm for shape detection, ensuring high accuracy in recognizing geometric shapes and determining their spatial properties. Experimental results demonstrate the robustness of the proposed method in various dynamic environments, highlighting its potential for enhancing object detection.

Keywords: mobile robotics, geometric primitives, LiDAR mapping, RANSAC, obstacle detection

1 Introduction

The advancements of dynamic environments in various fields of technology has necessitated the development of robust systems capable of recognizing and interpreting basic geometric shapes. The ability to map primitive shapes such as circles, rectangles, triangles, and pentagons, along with their intrinsic properties like center, radius, and size, is fundamental to the advancement of computational geometry and its applications. This project addresses the critical need for an adaptive framework that can efficiently recognize and catalog geometric shapes in real-time, which is essential for the seamless operation of dynamic systems.

Addressing this challenge is of paramount importance due to its wide array of applications. In autonomous navigation, for instance, accurately identifying and interpreting the shapes of objects and obstacles can greatly enhance path planning and collision avoidance mechanisms. Similarly, in space exploration, where environments are highly unpredictable and data is sparse, robust shape recognition can assist in tasks such as terrain mapping, object identification, and navigation. By improving the reliability and precision of shape detection in these contexts, we can enhance the operational efficiency and safety of autonomous systems, facilitating their integration into more complex and critical applications, towards a more automated and exploratory future.

This paper introduces an innovative approach using a mobile robot equipped with a LiDAR sensor to identify and map 2D primitives such as circles, rectangles, triangles, and pentagons. Our goal is to develop a method that not only enhances the robot's ability to discern these primitives accurately but also ensures computational efficiency, allowing for real-time processing and response in dynamic environments. The robot uses the Random Sample Consensus (RANSAC)[1] algorithm, optimized to recognize and parameterize each shape based on its geometric properties, which is critical for navigating through and interacting with the mapped environment.

The paper is organized as follows: Section 2 provides a review of existing technologies and methodologies employed in similar mapping tasks, emphasizing the use of RANSAC in robotic applications. Section 3 describes our methodology, detailing the use of LiDAR data, the segmentation process, and the step-by-step application of the RANSAC algorithm. Experimental results, including the algorithm's performance in terms of accuracy and computational efficiency, are presented in Section 4. Finally, Section 5 discusses the implications of our findings and suggests directions for future research in mobile robotics mapping.

2 Literature Review

Fitting geometric primitives to LiDAR point clouds is an emerging area of research, focused on adapting robust fitting algorithms to efficiently process data captured by LiDAR sensors. Several recent projects have made significant strides in this domain, each contributing unique approaches and insights.

Tóth and Hajder (2019) [2] present a method for robust surface fitting on spatial points measured by LiDAR devices. Their research focuses on plane, sphere, and cylinder detection, using modified RANSAC algorithms to enhance fitting accuracy and reduce computational time. The main contributions include a novel modification of RANSAC for robust plane fitting and similar modifications for sphere and cylinder fitting. They validate their methods using both synthesized and real-world indoor and outdoor data. While both their work and ours utilize RANSAC as a core component and aim to improve fitting accuracy and efficiency, there are notable differences. Tóth and Hajder's work covers a broader range of shapes compared to our focus on circles, rectangles, triangles, and pentagons. They apply a single RANSAC modification tailored to each shape, whereas our approach involves iteratively applying RANSAC, refining detection and parameter estimation starting with simpler shapes.

Xia et al. (2020)[3] introduce an efficient RANSAC algorithm tailored for 3D object detection in LiDAR point clouds. Their approach addresses the challenge of fitting 3D shapes such as planes, spheres, and cylinders by optimizing the computational efficiency and robustness of the RANSAC method. They emphasize reducing the number of iterations required by using a deterministic technique to select data points after an initial random sampling. Similar to our study, they aim to enhance the efficiency and robustness of RANSAC. However, their focus on 3D shapes and general object detection contrasts with our emphasis on

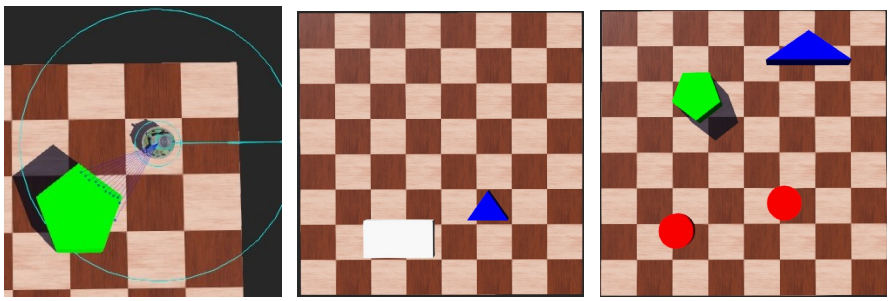
2D primitives and sequential detection in dynamic environments. Our method applies RANSAC iteratively, starting with simpler shapes and moving to more complex ones, differing from their approach of reducing iterations through deterministic sampling.

3 Algorithm and experimental validation

3.1 Experimental Setup

We utilized an e-puck robot equipped with a LiDAR sensor for our experiments. The specifications of the LiDAR sensor are as follows: the minimum range is 0.05 meters, and the maximum range is 0.3 meters. It has a 360-degree field of view (2π radians) and a horizontal resolution of 100 rays per 360 degrees. Figure 1a shows an example of LiDar sensor range and field of view. Additionally, the frequency interval at which the LiDAR performs a scan is every 100 ms.

To evaluate our approach, we created several maps containing different shapes. These maps were designed to test the robot's ability to detect and identify various geometric primitives under controlled conditions. We designed four different types of maps, including maps with only one shape, two shapes, three shapes, and four shapes. Each shape on the maps was randomly selected from a predefined list, which includes circles, rectangles, triangles, and pentagons. Shapes were generated with random sizes to introduce variability, ensuring that the detection algorithm could handle different scales and dimensions. All experiments were conducted on a board measuring 1 meter by 1 meter. The e-puck robot and the shapes were placed within this confined area, ensuring that the robot had to navigate and detect shapes within a well-defined space. Two examples of the generated maps are illustrated in Figures 1b and 1c



(a) Lidar Sensor (b) Map with 2 shapes (c) Map with 4 shapes

Fig. 1. Experimental setup

3.2 LiDAR Data Acquisition and Grid Representation

To begin, the LiDAR sensor attached to the e-puck robot performs a scan every time the robot takes a step. This frequent scanning allows for continuous updating of the spatial information. The data collected from the LiDAR sensor, which includes the distance measurements and angles, is used to update a grid. In this grid, each point's coordinates are adjusted to fit a predefined resolution of 0.001 meters. This adjustment ensures that the spatial data is uniform and easy to manage. For instance, a real-world point at coordinates (0.1, 0.1) would be represented as (100, 100) on the grid. This grid-based representation facilitates the efficient management and processing of point cloud data, allowing for accurate and reliable shape detection. An example of the final grid after the LiDAR has scanned the entire map can be seen in Figure 2 2a.

Robot Movement The movement of the e-puck robot within the experimental setup is a crucial aspect of our methodology. The robot's movement can be controlled in two ways: through user input or by teleporting it to predefined points. This flexibility allows us to conduct both manual and automated tests, ensuring that the algorithm is robust and versatile. User-controlled movement enables the testing of the algorithm in real-time scenarios, while predefined teleportation points allow for systematic and repeatable testing in controlled conditions. By varying the movement approach, we can evaluate the performance of our shape detection algorithm across a range of dynamic environments and motion patterns, ensuring its adaptability and effectiveness.

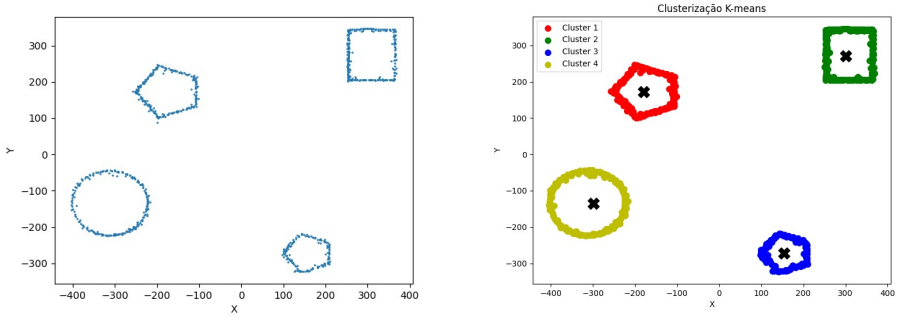
Point Segmentation Using K-Means Clustering After collecting the LiDAR data, the next step is to segment the points into distinct clusters corresponding to individual shapes. We utilize the K-Means[4] clustering algorithm for this purpose. K-Means is effective at grouping points based on their spatial proximity, which helps in isolating different geometric shapes present in the data. By applying K-Means, the points are separated into multiple lists, with each list representing a potential shape. This segmentation is essential for simplifying the subsequent shape detection process, allowing us to focus on smaller, more manageable subsets of data. Example of LIDAR data points visualization after applying K-means clustering in figure 2b.

Shape Detection with RANSAC With the segmented points, we proceed to the core of our methodology: applying the Random Sample Consensus (RANSAC) algorithm to each cluster. RANSAC is known for its robustness in fitting models to data sets with a significant amount of noise. For each cluster, RANSAC iteratively fits lines to the points, identifying inliers (points that fit the model) and outliers (points that do not). This process continues until the number of outliers is reduced to a predefined threshold, set as a percentage of the total number of points in the cluster (10% in our case). The RANSAC algorithm returns the number of iterations performed (indicating the number of lines detected) and a

list of inliers for each shape. This iterative refinement ensures that each detected shape does not interfere with the detection of subsequent shapes. If RANSAC detects a high number of lines (more than five), the shape is considered a circle. In such cases, we use the PyRANSAC library, which is specifically designed for fitting circular shapes. PyRANSAC efficiently handles the detection and parameter estimation for circles, ensuring accurate identification. This additional step for circle detection helps in differentiating complex shapes from simpler ones, enhancing the overall robustness of our method.

Shape Measurement Once the shapes have been detected using RANSAC, the next step is to measure their geometric properties. The number of lines and the inliers provided by RANSAC are used to calculate essential parameters, such as the center and radius for circles and the vertices for polygons. The measurements of the shapes are calculated through the coordinates of the vertices detected by RANSAC. The centroid of each shape is obtained by calculating the mean of all points present in the shape. This measurement step is crucial for accurately mapping the detected shapes and understanding their properties. In figures 3a, 3b, 3a and 3b we can see the center point and vertices of the different shapes.

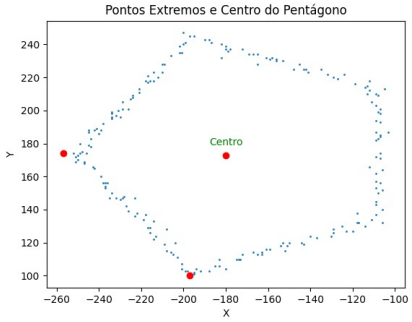
Conclusion By following these detailed steps, we have developed a robust and adaptable method for detecting and mapping primitive shapes in dynamic environments. This approach leverages the strengths of the RANSAC algorithm and K-Means clustering, combined with grid-based point storage and flexible robot movement, to achieve high accuracy and replicability in shape detection. The entire methodology described here can be visualized in the provided pseudocode 1 and explored in more detail available at https://github.com/campiao/cv-primitive-mapping.



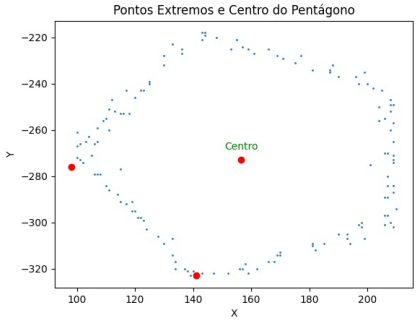
(a) Final Lidar Points

(b) Isolating different geometric shape with Kmeans

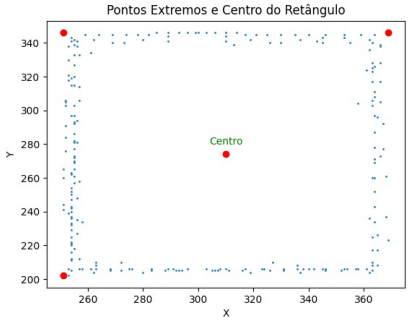
Fig. 2. LIDAR data points visualization before and after applying K-means clustering.



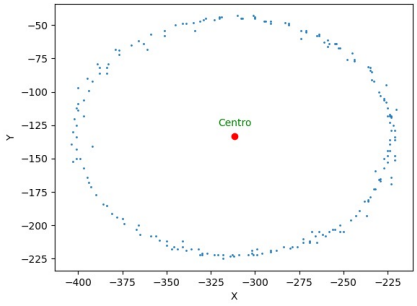
(a) First Pentagon



(b) Second Pentagon



(a) Rectangle



(b) Circle

Fig. 3. Shapes with center and vertices

Algorithm 1 LiDAR Data Acquisition and Shape Detection Algorithm

```
Input: LiDAR data from robot's environment
Output: Detected shapes with measurements
procedure CaptureData
    while robot is moving do
        Perform LiDAR scan
        for each point in scan do
            Convert real-world coordinates to grid coordinates
            Update grid with new point data
procedure SegmentData
    Collect all points from grid
    Apply K-Means Clustering to segment points into clusters
procedure DetectShapes
    for each cluster from SegmentData do
        Initialize RANSAC
        while inliers < 90% of points in cluster do
            Randomly select sample points
            Fit line to sample points
            Identify inliers and outliers
            Updated points that fit line
procedure MeasureShapes
    for each shape model from DetectShapes do
        if detected lines > 5 then
            Use PyRANSAC to fit a circle
            Calculate circle's center and radius
        else
            Calculate geometric properties and center
procedure Main
    CaptureData
    SegmentData
    DetectShapes
    MeasureShapes
    Output detected shapes and their measurements
```

3.3 Results

Randomly generated maps were used to evaluate the performance of the proposed algorithms. Specifically, ten maps each containing two, three, and four primitives were created. Additionally, four maps were generated with individual primitives to further test the algorithm under varied conditions.

Two test cases were employed to assess the algorithm's robustness: one using automatic movement and the other using manual movement. The average values obtained from the ten randomly generated maps for each scenario are presented in Tables 1, 2, 3, and 4. These tables also include the average time required to execute the scripts.

The execution time for the automatic movement scenario remained constant across tests. However, in the manual movement scenario, the execution time increased proportionally with the number of primitives in the map, though it never exceeded one minute in our test cases.

To evaluate performance more comprehensively, we adopted the following weighted accuracy formula:

$$\text{Weighted Accuracy} = 0.4\times\text{Shape Accuracy}+0.3\times\text{Center Accuracy}+0.3\times\text{Dimension Accuracy} \quad (1)$$

Where Shape Accuracy represents the correct identification of the shape, Center Accuracy refers to the precision in locating the center of the shape, and Dimension Accuracy pertains to the accuracy of other dimensional measurements (e.g., radius in the case of circles, side length for squares).

Greater emphasis was placed on Shape Accuracy (40%) because correctly identifying the shape is crucial for the accuracy of other metrics. Both the center and dimensional measures were weighted at 30% each. Additionally, each metric was individually analyzed by counting the instances of correct identification and dividing by the total number of measurements taken.

These methods allowed us to present a detailed and quantifiable analysis of the algorithm's performance across various scenarios.

Table 1. Accuracy results and time spent running for the algorithm running automatically on 1 shape maps

Shape	**Total accuracy**	**Shape type accuracy**	**Center position accuracy**	**Shape measurements accuracy**	**Time spent running (s)**
Circle	1.0	1.0	1.0	1.0	10.03
Triangle	0.85	1.0	0.50	1.0	9.43
Rectangle	0.85	1.0	1.0	0.50	9.48
Pentagon	0.70	1.0	1.0	0.0	9.70
Average	0.85	1.0	0.875	0.63	9.66

Table 2. Average accuracy results and time spent running for the algorithm running automatically on multiple shapes maps

Shape	**Total accuracy**	**Shape type accuracy**	**Center position accuracy**	**Shape measurements accuracy**	**Time spent running (s)**
2 shapes	0.83	1.0	0.85	0.57	10.37
3 shapes	0.83	1.0	0.87	0.57	10.70
4 shapes	0.87	1.0	0.92	0.67	12.03

Table 3. Accuracy results and time spent running for the algorithm running mannualy on 1 shape maps

Shape	**Total accuracy**	**Shape type accuracy**	**Center position accuracy**	**Shape measurements accuracy**	**Time spent running (s)**
Circle	1.0	1.0	1.0	1.0	13.50
Triangle	0.55	1.0	0.5	0.0	12.77
Rectangle	0.85	1.0	1.0	0.50	14.18
Pentagon	0.70	1.0	1.0	0.0	13.45
Average	0.77	1.0	0.875	0.38	13.48

Table 4. Average accuracy results and time spent running for the algorithm running mannualy on multiple shapes maps

Shape	**Total accuracy**	**Shape type accuracy**	**Center position accuracy**	**Shape measurements accuracy**	**Time spent running (s)**
2 shapes	0.72	1.0	0.70	0.38	28.34
3 shapes	0.67	0.86	0.68	0.40	39.25
4 shapes	0.80	1.0	0.83	0.50	58.25

The analysis of the results obtained from the experimental evaluations reveals several key insights into the performance of our detection algorithms. Firstly, the correct shape and number of shapes are almost always accurately identified, showcasing the efficacy of our shape detection methodology. Furthermore, the position of the center is also determined with high accuracy in most cases, indicating reliable spatial resolution capabilities of our system.

However, the measurement of shape dimensions presents our most challenging results. This discrepancy is primarily due to inaccuracies inherent in LiDAR readings, which occasionally produce erroneous data points, leading to deviations in precise dimension calculation.

It is also noted that the figures are almost always detected correctly, with only a few errors in determining the two properties mentioned above.

When comparing the two test cases—automatic versus manual movement—a significant difference is observed in both the time efficiency and accuracy of data collection. The automatic process generally achieves better accuracy as the data collection is more uniform and does not depend on the user-defined path, which can vary significantly in the manual mode. This uniformity in automatic data collection ensures that the algorithm performs consistently under controlled conditions.

4 Conclusions and Future Work

This study presented a comprehensive approach to detecting and mapping primitive shapes using LiDAR data in a dynamic environment. The implementation of our algorithm across various test scenarios has demonstrated robust capabilities in accurately identifying the number and type of geometric shapes, as well as precisely locating their centers. Our results underscore the effectiveness of the K-Means clustering and RANSAC algorithms in handling spatial data with high reliability, especially when supported by automated data acquisition processes.

The evaluation revealed that while the shape detection and center positioning achieved high levels of accuracy, measuring the dimensions of these shapes posed some challenges, primarily due to the inherent limitations of LiDAR technology in capturing exact details under certain conditions. Despite these challenges, the overall performance of the automated system proved superior in terms of both accuracy and efficiency compared to manual data collection methods.

The distinction between automated and manual movement highlighted the importance of consistent data collection techniques in achieving optimal results. The automated approach facilitated a more uniform data collection process, which was crucial in minimizing errors and enhancing the reliability of the detected shapes.

In conclusion, the methodologies employed in this study have proven effective in the context of LiDAR-based shape detection and mapping. The high accuracy in shape and center detection confirms the potential of these techniques for applications in automated navigation and robotic perception, where precise

environmental understanding is critical. This work establishes a strong foundation for utilizing advanced LiDAR processing techniques in practical, real-world applications in robotics and related fields.

4.1 Future Work

The promising results obtained from this study enables us to explore future research in the field of 3D shape detection and environmental mapping. First, expanding the capabilities of our existing algorithms to integrate camera data alongside LiDAR inputs could significantly enhance the system's ability to perceive and interpret complex environments. Cameras, which capture detailed texture and color information, can complement LiDAR sensors that provide precise distance measurements. Combining these two types of data through sensor fusion techniques could lead to more accurate and robust detection of 3D shapes, particularly in visually complex scenarios.

Moreover, the deployment of multi-layer LiDAR sensors offers another promising direction. These sensors can capture a more comprehensive 3D view of the environment by providing multiple scanning layers, which could improve the algorithm's ability to discern between overlapping or closely situated objects. The use of advanced point cloud processing algorithms that can effectively manage and interpret these denser data sets will be crucial in maximizing the benefits of multi-layer LiDAR technology.

Another important area of future research is the detection of composite shapes. Current methodologies excel at identifying simple geometric figures, but real-world environments often contain objects that are composites of several basic shapes. Developing algorithms capable of decomposing such composite shapes into their constituent parts and understanding their spatial relationships would significantly enhance the system's applicability in real-world scenarios. This could involve advancements in machine learning techniques that are trained on a diverse set of environments and object configurations.

These future research directions not only promise to improve the accuracy and reliability of shape detection algorithms but also extend their applicability to a broader range of practical and industrial applications. The integration of these technologies could ultimately lead to more intelligent and adaptable robotic systems capable of operating autonomously in a variety of complex environments.

References

[1] Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM 24(6), 381–395 (jun 1981), https://doi.org/10.1145/358669.358692

[2] Toth, T., Hajder, L.: Robust Fitting of Geometric Primitives on LiDAR Data. VISIGRAPP pp. 1–8 (2019)

[3] et al, S.X.: Geometric Primitives in LiDAR Point Clouds: A Review pp. 1–23 (2020)
[4] Jin, X., Han, J.: K-Means Clustering, pp. 563–564. Springer US, Boston, MA (2010), https://doi.org/10.1007/978-0-387-30164-8_425