Feature-Based Object Detection using Multiple 2-D Laser Range Finders

Nicholas Johnson, Dr. Carl Crane
Center for Intelligent Machines and Robotics
300 MAE Building B
Gainesville, FL 32611
1-352-392-9461
nichmj@ufl.edu, ccrane@ufl.edu

Dr. A. Antonio Arroyo, Dr. Eric Schwartz
Machine Intelligence Laboratory
325 MAE Building B
Gainesville, FL 32611
1-352-392-6605
arroyo@mil.ufl.edu, ems@mil.ufl.edu

ABSTRACT

Laser Range Finders (LRFs) are popular sensors used in object detection applications such as Simultaneous Localization and Mapping (SLAM) and the Detection and Tracking of Moving Objects (DATMO) due to their high range accuracy, low cost and low processing demands. However, many papers use a single LRF when addressing the SLAM and DATMO problems. The use of multiple LRFs can improve a robot's understanding of the world and provide data that a single LRF cannot due to occlusion or range limitations. Although some work has been done using multiple LRFs, grid-based approaches are used to represent the environment around the robot. Grid-based representations require greater memory and the use of image processing techniques to identify individual objects from the environment, which can be slow and demands additional processing power. Feature-based world representations can usually represent an object more concisely and therefore require less storage space. Also, no additional processing is required to identify individual objects since they are already represented separately. The work presented here investigates methods for using multiple LRFs in a feature-based object detection system. Three fusion approaches were considered. First, a point fusion approach in which all the data points from both LRFs are combined before the object detection processing is performed. Second, an object fusion approach in which objects are detected from both LRFs separately, and objects that lie within overlapping regions are combined. Finally, a shared object approach, in which each LRF works with the same shared list of objects during the object detection process. Discussions on each approach are provided. The shared object approach was deemed the most suitable and implemented in a feature-based object detection system. All objects in the environment were known to be static and line segments were used to represent the detected objects. Two LRFs that were assumed to scan along the same plane had varying positions on the robotic platform to provide a region of overlap. The chosen approach was shown to be successful and effective but was susceptible to the accuracy of the robot position. The incorporation of a SLAM system would greatly improve the results. Experimental results and conclusions are introduced and discussed in the paper.

Keywords

Multiple Laser Range Finders, LADAR, Feature Based Object Detection, Sensor Fusion.

1. INTRODUCTION

When working with autonomous vehicles, SLAM [1] and DATMO [2, 3] are two areas that have generated a lot of interest. Although there are many sensors that can be used in these applications, LRFs are the most common due to their high accuracy of the range data, ease of use, low cost and processing speed [4]. However, there are very few LRFs that can provide a 360 degree field of view and those that do can be very expensive and require mounting above the vehicle, which is often problematic. One possible solution to this problem is the use of multiple LRFs with overlapping regions in order to increase the field of view [5]. Although a reasonable approach, questions about how to fuse the point data generated from two different LRFs arise, especially when considering that they are running independently and may be mounted at different locations on the robot. The work done here introduces three possible approaches to the data fusion question and evaluates their suitability. When devising the approaches discussed, there were three main requirements:

1. The object detection should be feature-based and not grid-based.
2. In order to maintain the advantages of LRFs, the use of image processing techniques was deemed unfavorable.
3. There should be minimal changes between using a single LRF and using multiple LRFs.

Figure 1. The Urban Navigator.

2. TEST SETUP

The University of Florida's 2007 DARPA Urban Challenge vehicle was used as the test vehicle for the developed system. The Urban Navigator (Figure 1) is a modified 2006 Toyota Highlander Hybrid SUV equipped with numerous LRFs, cameras, and a positioning system which combines differential GPS with an Inertial Measurement Unit (IMU) and wheel encoders. Data from two SICK LD-LRS1000 laser measurement units mounted to the hood of the vehicle was used to test the various fusion approaches (Figure 2). Each LRF was configured to scan a 270 degree region, with a 0.25 degree angular resolution at 10Hz. Figure 3 shows the scan regions for each LRF along with the overlapping area. All testing was done with the vehicle remaining stationary and the global position estimate coming from the positioning system. Only static objects were considered.

Figure 2. The position of the LRFs used for testing (marked as Passenger and Driver Long-Range LADAR).

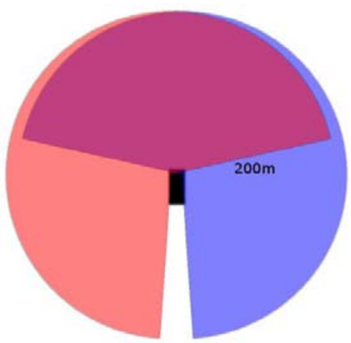

Figure 3. Scan regions for the two LD-LRS100 laser range finders.

3. METHODOLOGY

In order to develop the LRF fusion approaches, an object detection system using a single LRF was first developed. Figure 4 shows the general steps for the detection system implemented. The clustering and line segment steps are used to detect objects in the current scan while the matching and resolution steps are used to track the object over time and ensure that an object is consistently identified as the same object. An object is considered to be consistently identified if its identification number does not change over time due to sensor noise.

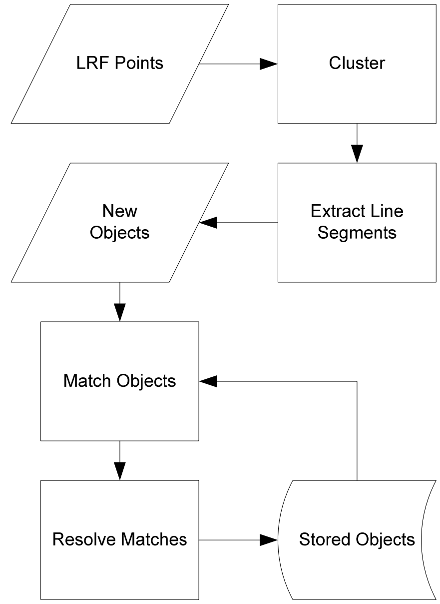

Figure 4. Flowchart outlining the object detection and tracking system.

3.1 Object Detection

There are a number of well established methods and techniques for using LRFs in object detection and feature extraction. Line segments are a popular feature as many static objects in structured (indoor) and semi-structured (urban / city) environments have a regular shape (rooms, corridors, buildings, cars, etc). Therefore, the established adaptive breakpoint detector and the iterative end-point fit methods [6-8] were used for clustering and line feature extraction, respectively.

Figure 5. Scan points (red) overlaid on a satellite image of the test area. The vehicle position is shown by the gray box and blue lines.

One goal of the developed system is to represent a singular physical object with a singular representation, that is, a building should be identified as a single object and not as a series of walls. In Figure 5 the white building in the middle (in front of the vehicle) should be identified as a single object. However, due to sensor noise and occlusion, the clustering algorithm, cannot cluster the points together as a single object. Figure 6 shows how the white building shown in Figure 5 is divided into 4 different clusters. The matching and resolution steps as well as the combination of multiple LRFs can lessen this problem.

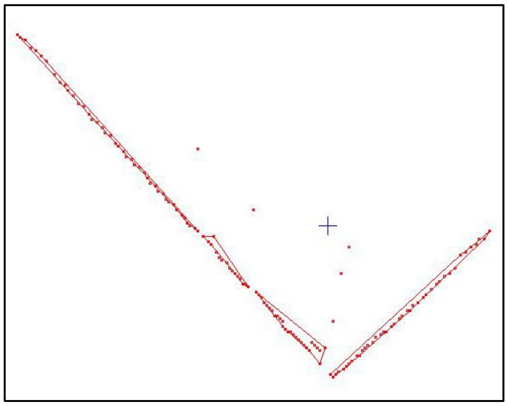

Figure 6. Clusters generated from scan points of the white building in the middle of the image in Figure 5.

3.2 Object Matching

Sensor noise can cause the clusters generated and the final object shape to vary between scans (Figure 7). Therefore, in order to maintain a consistent and stable understanding of the objects in the environment, the newly detected objects and the previously detected objects need to be matched and any differences need to be resolved.

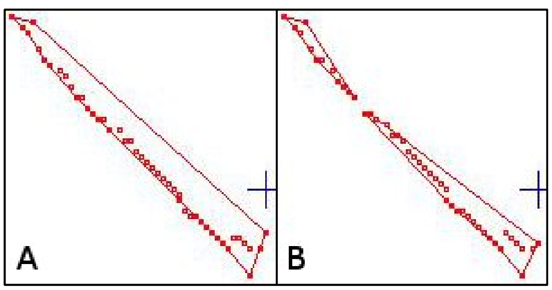

Figure 7. Example of different clustering on points from the same object due to sensor noise. A) t = 0. B) t = 1

In order to match the objects between scans, an enclosure around each line segment is generated based on the line break distance used during the line fit (Figure 8). Every new object detected in the current scan is compared to every previously detected object and if the enclosures overlap they are considered to be matching objects. There are four different match scenarios that can occur after comparing the objects:

1. One-to-one match: a new object matches one and only one old object and the one old object only matches the one new object.
2. One-to-many match: a new object matches multiple old objects but the multiple old objects only match the one new object or vice versa.
3. One-to-many-plus match: a new object matches multiple old objects and any of the old objects match different new objects or vice versa. However, none of the new objects overlap and none of the old objects overlap.
4. Many-to-many match: a new object matches multiple old objects and any of the old objects match different new objects or vice versa. The new objects can overlap and the old objects can overlap.

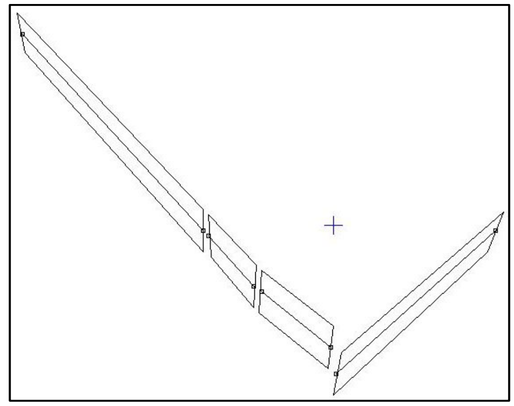

Figure 8. Detected objects with enclosures around each line segment.

3.3 Object Resolution

After the objects have been matched, they need to be resolved and the previous objects updated based on the new scan. An important requirement of the object detection system is stability. If the shape of an object is constantly changing, the navigation and control elements could become unstable as the robot attempts to avoid the object. Also, noise is introduced into any higher level information extraction, such as velocity estimation. Therefore, when the line segments between a previously detected object and a newly detected object vary, an averaging scheme is used to stabilize the objects.

In one-to-one matches, resolution is simple; the new object is the same as the old object. The overlapping regions in the old object are updated based on the new object. However, when considering the one-to-many and one-to-many-plus match scenarios, there are two possibilities: either two objects were incorrectly clustered as a single object or a single object was incorrectly detected as two objects. The approach taken in this research was to assume that the objects were incorrectly detected as separate objects and were always merged. However, this assumption may not hold true in all cases especially when the robot starts to move or there are dynamic elements in the environment. When using a single LRF, many-to-many matches do not occur as it is not possible for new or old objects to overlap based on the sensor properties. However, when using multiple LRFs, many-to-many matches may occur based on the fusion approach used and the resolution scheme will be discussed as needed.

4. FUSION APPROACHES

Three approaches for fusing the LRF data were evaluated: a point fusion approach, an object fusion approach, and a shared object approach. Each approach is discussed below.

4.1 Point Fusion Approach

In the point fusion approach, data from both LRFs are treated as if they are generated by a single (combined) LRF, which is positioned over the vehicle's reference frame. Therefore, the methodology employed for a single LRF does not need to be modified when using multiple LRFs. Data from each LRF was assumed to be taken at the same time. As the vehicle was stationary, this assumption was reasonable, however, at speed there could be a large discrepancy between points which would need to be handled. Although this is a simple approach that initially appeared to work well, it suffered from a number of problems.

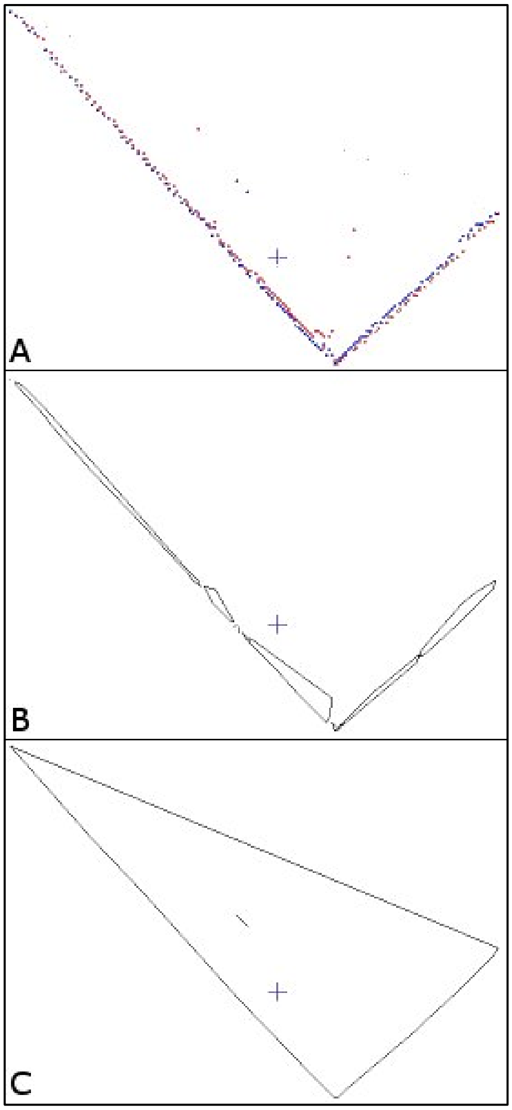

Figure 9. A) Scan points from the two LRFs. B) The clusters generated using the unmodified algorithm. C) The clusters generated using the modified algorithm.

The first problem is that free space information which is inherent to the point data is lost when the points are transformed to the shared frame. After transformation, it is no longer valid to assume that the straight line from the combined LRF position to the scan point is free space. The second problem is that the standard clustering algorithm fails. The algorithm exploits the property of the LRF data which guarantees that a point will not exist behind another point within the scanned region due to occlusion (point occlusion property). When the data from both LRFs are combined, this property is no longer valid since each LRF has a slightly different viewing angle. Although, the clustering algorithm can be modified to produce reasonable results by comparing each point to every detected cluster, it increases the required processing load. Also, clusters can now overlap (which again violates the occlusion property of the LRF) which might need to be handled during the matching stage. Figure 9 shows the clusters generated by the unmodified algorithm and modified algorithm on the combined LRF.

The modified algorithm produces reasonable clusters; however, the line fit algorithm now produces undesirable results and leads to object "shading" due to the absence of the point occlusion property. Figure 10 shows how the line segments representing the object are not as smooth (compared to the individual LRFs) after fitting a line through the points. The "shading" effect is caused by the error between the LRFs.

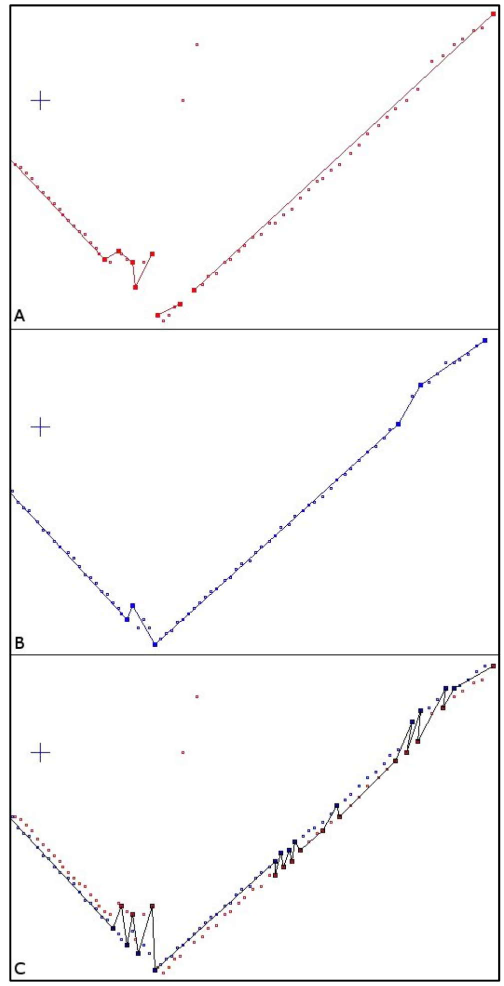

Figure 10. A) Line segments extracted using points from the driver LRF. B) Line segments extracted using points from the passenger LRF. C) Line segments extracted using points from the combined LRF.

Other line extraction techniques may work better and image processing techniques such as the Hough Transform [7, 9] would probably work well. However, due to the previously mentioned problems with this approach and the system requirements, it was deemed inappropriate.

4.2 Object Fusion Approach

In the object fusion approach, objects are detected from each LRF separately and then merged into a single object for use in the matching step. As detection and feature extraction was performed on each LRF separately the clustering and line fitting algorithms were used as-is, i.e., without modification. In order to merge the objects between the LRFs, additional matching and object resolution must be done. The objects are represented in the global frame and are assumed to be generated at the same time. Figure 11 shows the updated flowchart for this approach.

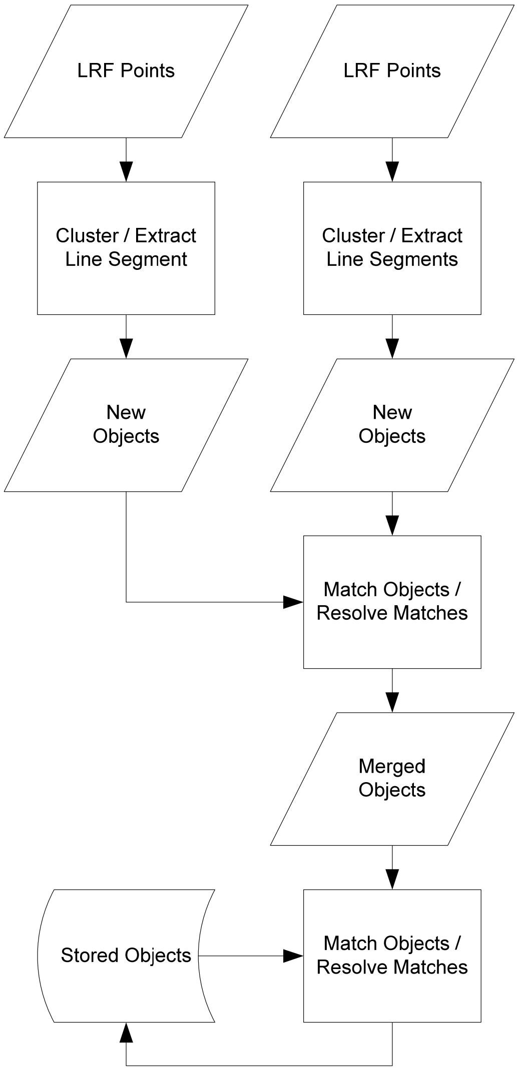

Figure 11. Updated flowchart for the object fusion approach.

Although this approach overcomes the problems of clustering and line extraction seen with the point fusion approach, it has two main problems. First, the additional matching and resolution step increases the processing requirements significantly. The matching step is the most processor intensive portion of the process due to the need to compare every object against every other object to find all possible matches. Although, the additional matching step can be eliminated by not merging the objects from each LRF, the removal of the merging step introduces the problem of dealing with the occurrence of many-to-match matches. When attempting to resolve objects in a many-to-many match there are two problems: the algorithm needed to resolve the objects is very complex and difficult to implement, and a method for updating an old object when there are two overlapping new objects that match is difficult to define. The next problem with the object fusion approach is that object merging between LRFs is not a simple task. After the two objects are generated, it becomes difficult to correctly determine which sections of the object should be merged, since the objects are represented in the global frame and the inherent free-space information obtained by a scan point is lost when the points are transformed. This approach was also deemed inappropriate due to the outlined problems, but lead to the development of the final approach discussed below.

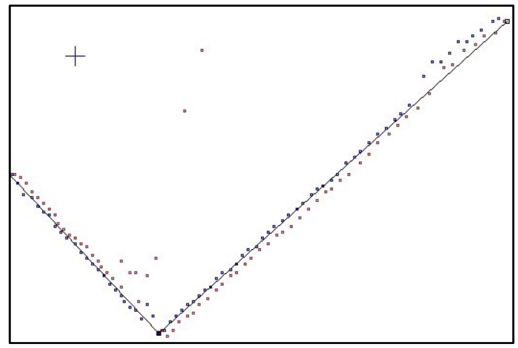

Figure 12. Shared object approach: Good results generated.

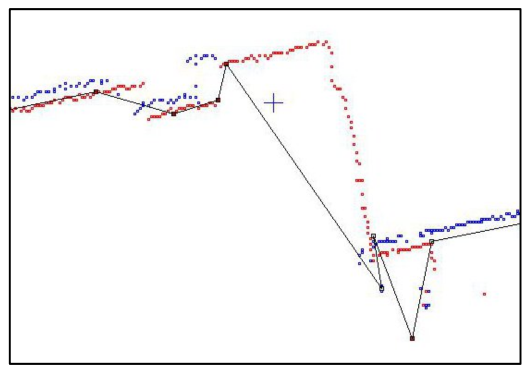

Figure 13. Shared object approach: Incorrect results generated due to the large occluded sections between LRFs.

4.3 Shared Object Approach

In the shared object approach, each LRF is treated independently but processing is run sequentially when new data from an LRF is received. Objects can only be updated by data from one LRF at a time and therefore, there is no need to assume that the data from each LRF is received at the same time. In fact, the LRFs can run at different rates and at different resolutions. All objects are stored in the global frame and are first converted to the LRF frame before any processing is done. Although, the transformation between frames adds additional processing power, it is less intensive than performing the matching and resolution step required in the object fusion approach. Also, since only the conversion from the global frame to the LRF frame and back needs to be known for any LRF, the approach is fairly general. One major advantage of this approach is that almost no changes have to be made to the existing, single LRF process, since each LRF acts as though it is the only one working with the objects. The output from the system was shown to work well (Figure 12), but suffers when there are large sections of an object that are observable by one LRF but not the other due to occlusion (Figure 13).

5. CONCLUSIONS

The shared object approach produced the best results based on the research requirements. The point fusion approach suffered from a break down in the common algorithms used for processing LRFs and produced inadequate object models, while the object fusion approach added additional complexity and processing load. The shared object fusion approach closely follows the single LRF methodology which does not introduce much additional complexity or processor load, while still producing good results. Also, this method seems to lend itself to the fusion of data from different platforms and does not require the LRFs to be on the same vehicle. However, additional work is required to deal with issues when the LRFs do not have the same observable view. Also, the system suffers from error in the positioning system. If the estimate generated from the position system changes greatly between time steps, the object matching and resolution can become inconsistent and unstable. However, this is a problem that is also present in the single LRF detection system. The incorporation of SLAM to correct position errors would greatly improve the system.

6. ACKNOWLEDGEMENTS

The authors would like to acknowledge the support of the Air Force Research Laboratory at Tyndall Air Force Base, FL.

7. REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," Robotics & Automation Magazine, IEEE, vol. 13, pp. 99-110, 2006.

[2] M. Lindstrom and J. O. Eklundh, "Detecting and tracking moving objects from a mobile platform using a laser range scanner," Maui, HI, United states, 2001, pp. 1364-1369.

[3] Y. Jin-Xia, C. Zi-Xing, and D. Zhuo-Hua, "Detection and tracking of moving object with a mobile robot using laser scanner," in Machine Learning and Cybernetics, 2008 International Conference on, 2008, pp. 1947-1952.

[4] H. Zhao, X. W. Shao, K. Katabira, and R. Shibasaki, "Joint Tracking and Classification of Moving Objects at Intersection Using a Single-Row Laser Range Scanner," in Intelligent Transportation Systems Conference, 2006. ITSC '06. IEEE, 2006, pp. 287-294.

[5] K. C. J. Dietmayer, J. Sparbert, and D. Streller, "Model based Object Classification and Object Tracking in Traffic Scenes from Range Images," Proceedings of IV IEEE Intelligent Vehicles Symposium, Tokyo, 2001.

[6] G. A. Borges and M.-J. Aldon, "Line extraction in 2D range images for mobile robotics," Journal of Intelligent and Robotic Systems: Theory and Applications, vol. 40, pp. 267-297, 2004.

[7] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2D laser rangefinder for indoor mobile robotics," in Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on, 2005, pp. 1929-1934.

[8] C. Premebida and U. Nunes, "Segmentation and geometric primitives extraction from 2D laser range data for mobile robot applications," 2005.

[9] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," Taipei, Taiwan, 2003, pp. 1304-1311.