Extracting Points Features from Laser Rangefinder Data Based on Hough Transform

Novel Certad[1] and Carlos Mastalli[1] and José Cappelletto[1] and Juan C. Grieco[1]

***Abstract*— This paper describes a novel feature extraction method for laser rangefinder data. Extracted features correspond to real and virtual corners of the scanned scene. The method is based on the Hough Transform (HT) for line extraction, where the intersecting points of these lines are considered as features. This work highlights the use of the HT outside of image applications, and presents a new filtering algorithm that reduces false positive in line detection by the HT based method. The developed method was tested under various simulated benchmarks in order to compare the performance as a function of correctness, uncertainty, execution time and other parameters. Also, a real data benchmark was included in the tests. Finally, a simulation of EKF-SLAM was performed to demonstrate the functionality of the developed method in more complex tasks.**

I. INTRODUCTION

Nowadays there are multiple areas of development in robotics. However, there are common problems like mapping, navigation, path planning, localization, and so on that have been solved in different ways. Many of these problems have been solved by feature-based algorithms like EKF (Extended Kalman Filter) for local localization and SLAM (Simultaneous localization and Mapping) [1]. The feature-based algorithms can't handle large amounts of raw data. That is why it is necessary to include a intermediary step between the sensor data and the feature-based algorithms. That step is called feature extraction (or landmarks extraction or references extraction).

The extraction and processing of spatial features (landmarks) from the data supplied by these sensors has become a sub-topic of great importance for the navigation of mobile robots. The extracted features must be stationary and re-observables, in order to achieve the convergence of the main algorithm (which could be a SLAM algorithm as shown shortly). In this paper we consider only point-features (also known as geometric references [2]) ie, those that are determined by its two coordinates in the plane and are equals regardless of the orientation.

Commonly, the most used sensors are cameras, ultrasounds and laser rangefinders. The latter one is used in the course of this work.

The method developed in this work, is a corner detector. A real corner occurs when the laser beam scans two different surfaces that intersect with nonzero angle. The method also detects virtual corners, these are obtained by prolonging a surface (wall, door, etc.) linearly until it intersects another one [3].

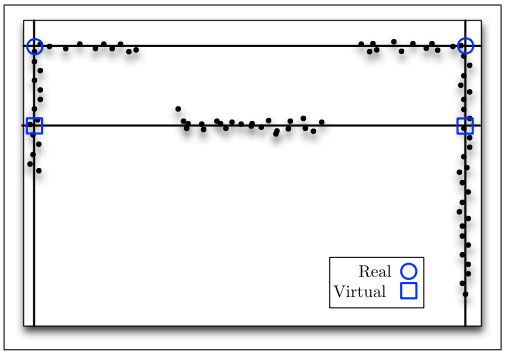

Fig. 1. Real and virtual corners.

Fig. 1 shows a scan example. The black dots represent raw laser measurements. Four lines are presents which produce four intersection points. The circle-marked ones are real corners and the others are virtual corners.

In 2010, Werneck et al. used a video taken by hand by an operator moving in a straight line down a hallway [4]. They used the Hough Transform to extract lines, which were then used to construct a map of the hallway. That same year, Santana et al. implemented EKF-SLAM using as references pre-existing lines on the floor that were extracted using the Hough Transform and monocular vision [5]. In [6] and [7], the Hough Transform is used to process ultrasound data.

Y. Li and E. Olson introduced in 2010, using techniques of image processing to extract references from laser data [8]. They passed the laser data by a rasterization process to convert it into an image.Then they used the Hough Trasform and the Kanade-Tomasi corner detector [9].

As seen, the Hough Transform has been widely used in image processing for robotics. However we propose to use laser data directly without converting the planar view of measurements in a 2D image.

In the next section will be explained the developed method.

II. REFERENCE EXTRACTION

As illustrated in Fig. 2, the references extraction algorithm based on Hough transform consisted of 5 blocks.

The first block applies the Hough transform to a full scan of the laser rangefinder ($\boldsymbol{\rho}_k$) obtaining as an output the accumulator array (Acc). The next block is the peak detector, which searches for local maximal in the accumulator array.

*This work was supported by the Research and Development Deanship (DID) of Simon Bolivar University.

[1]All authors are with Mechatronics Research Group, Simon Bolivar University, Baruta 89000, Venezuela ncertad@usb.ve

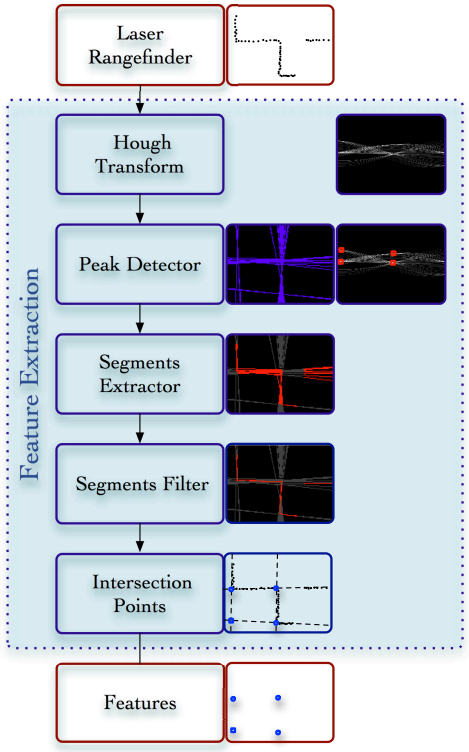

Fig. 2. Block diagram of the developed feature extraction algorithm.

It sets a threshold, above which must be the peak values to be detected.

The third block, compares each of the extracted lines with the scan measurements. This in order to determine which portion (line segment) of each line correspond to actual data, and how many measurements are associated with each segment, along with the error for each.The fourth block is responsible for removing the segments (or lines) that were detected but whose error or number of points does not make them reliable to be taken as real segments.

Finally, the fifth block, is responsible for finding the intersection points between all lines that survive the previous block, since those points are the wanted features.

The following describes in detail each of the blocks of the reference extraction algorithm.

A. Hough Transform And Peak Detector

The Hough Transform was initially introduced as a technique for extracting straight lines in images [10]. The straight lines are parametrized by two values:

- θ: the angle of its normal.
- d: its algebraic distance from the origin.

These parameters are illustrated in the Fig. 3, being Eq. 1 the equation of the line in terms of them.

$$d = x_p\, cos(\theta) + y_p\, sin(\theta) \tag{1}$$

The laser measurements in the $x - y$ plane are transformed to a $d - \theta$ plane. This plane is represented by a two-dimensional array called accumulator (Acc). Each pair

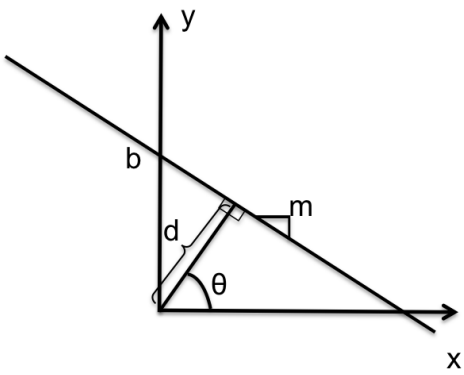

Fig. 3. Line parameters. The slope-intercept form equation uses m (slope of the line) and b (y-intercept of the line). Hough Transform uses the $d-\theta$ parameters.

of coordinates (d, θ) in the accumulator corresponds to a possible line in the $x - y$ plane and each point in the $x - y$ plane corresponds to a sinusoidal curve in $d - \theta$ plane.

One dimension of the accumulator is the quantized angle θ and the other dimension is the quantized distance d. The resolution of the grid is determined by Δd and $\Delta\theta$ values.

The larger accumulator values indicate the parameters (d, θ) of the lines in $x-y$ plane. As the peak values decrease, this may correspond to noisy spurious peaks instead of real lines. For this reason is created a threshold ($PeakTh$) that determines the minimum value that should take a peak to be considered such. We fit the 15% of the maximum value of Acc as the minimum value.

Additionally a minimum threshold ($MinTh$) was set, so that if the 15% of the maximum value of the accumulator does not exceed the minimum threshold is the latter that is used to detect peaks. The equation 2 shown how is determined the threshold.

$$PeakTh = max(MinTh, round(0.15 \times max(Acc))) \tag{2}$$

Logically, accumulator values found in the neighborhood of a detected peak also could exceed the threshold ($PeakTh$) and be considered maximal peaks. To ensure that the detected peaks are truly local maximal, whenever a peak is detected, is created a neighborhood around of the peak, in which is ignored the value of the accumulator to the following detections ($Neighborhood = \begin{bmatrix}\Delta V_d & \Delta V_\theta\end{bmatrix}$).

The size of the neighborhood is given by a quantity ΔV_d of cells in dimension d of the array and an amount ΔV_θ of cells in dimension θ. Both quantities must be integers and odd.

Additionally, it sets a parameter for limiting the number of peaks (lines) which detects the algorithm (N_{max}).

B. Segments Extractor

A line has no endpoints, this block extracts bounded segments from the previous detected lines. Each segment corresponds to a scan performed by the laser to a flat surface. each line can have one or more associated segments.

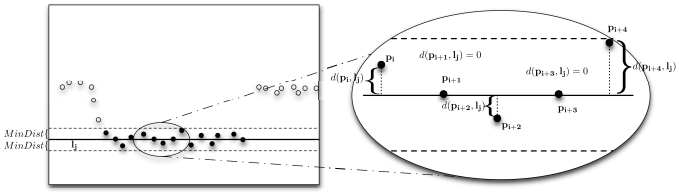

Fig. 4. black dots are those associated with line l_j. White dots are the unassociated ones.

For each line (l_j) extracted by the Hough Transform, the algorithm calculates the perpendicular distance from said line to each of the points corresponding to measurements of the laser ($d(P_i, l_j)$). Those points whose distance ($d(P_i, l_j)$) is lower than a certain threshold ($MinDist$) are considered points associated with that line. The Fig. 4 shows a series of points representing measurements of the laser. The black dots correspond to those that are within the threshold ($MinDist$) and are considered part of l_j line.

Each portion of a line that is associated to laser measurements is considered a segment. Each segment is determined by its endpoint coordinates.

C. Segments Filter

Due to the noise of the measurements and the inherent operation Hough transform, it often happens that for a real segment, two or more lines are detected. These detected lines have very little variations in the parameters d and θ. This block has the function of eliminating or reducing these lines.

First are removed the segments whose number of associated points is below a minimum. Then are removed those whose estimated error exceeds a threshold ($MaxError$).

Then the segments are revised (from different lines each) seeking for those whose angle of intersection is too small to consider they are distinct segments. In this case is deleted the segment that has higher estimation error or the smallest number of associated points.

Later we review all the lines extracted and are eliminated those who don't have any associated segment.

D. Intersection Points

From the previous block is received an array of the parameters of the detected lines that were not removed by the filter. Then we search for the intersection points of all lines that are not parallel. These are the point-features extracted by the developed method

It is possibly that some lines intersect at points very far to the order of magnitude of the field of view of the laser rangefinder. For example in the case of almost parallel lines.In order to remove these points, we fit a limit distance of 20m from the laser sensor to any detected point-feature.

III. IMPLEMENTATION

The algorithms described in section II were implemented using MATLAB 2009b, running on a laptop with Core i5@2.4 GHz and 4 GB of memory. Two different test benches were used. The first one was made with simulated data and the second with real data.

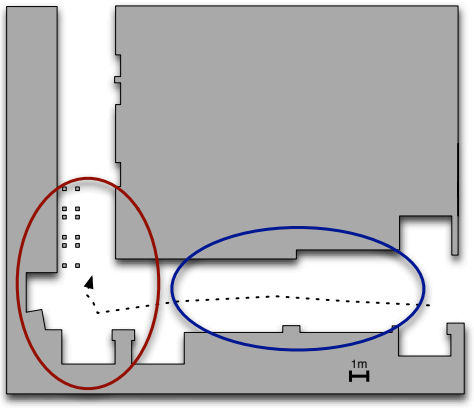

Fig. 5. L-shaped hallway. It is divided in two zones. In Blue zone predominate two parallel lines so there aren't almost any intersection point. In red zone there are several oblique lines so there are several intersection points

A. Simulation

For the simulation, we use Webots 6. It allows the use of an ODE physics engine that simulates the dynamic behavior (force interaction, gravity, friction) real time, it can also simulate a range finder laser with the statistical error associated in the measurements while considering the conditions of color of the object surfaces. A robotic platform of size, shape and other properties similar to MobileRobotics Inc.'s Amigobot was implemented, as well as a laser rangefinder similar to the SICK-LMS200 model, with the following features:

- $\sigma_r = 0.005$, Standard deviation of range measures provided by the Laser rangefinder.
- $\Delta\phi = 1°$, Angular resolution of the laser rangefinder.
- $ROV = 10m$, Range of view.
- $AOV = 180°$, Angle of view.

In the simulation were obtained over 300 laser scans to test the feature extraction method. Additionally, was simulated an EKF-SLAM algorithm to probe the functionality of the developed method in a more complicated navigation task

B. Real data

To test the algorithm with real data, we used a Hokuyo's laser rangefinder. The URG-04LX-UG01 model, with the following features:

- $\sigma_r = 0.005$, Standard deviation of range measures provided by the Laser range finder.
- $\Delta\phi = 0.33°$, angular resolution of the laser range finder.
- $ROV = 4m$, Range of view.
- $AOV = 240°$, Angle of view. (on practice was limited to $180°$)

The laser was moved through a L-shaped hallway, which is shown in Fig. 5.

C. Parameters

The most important parameters for the developed method are those for the Hough Transform and the peak detector.

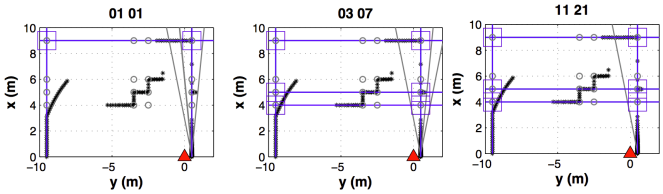

Fig. 6. A scene with detected features. Black points are raw measurements. Gray lines are those that was filtered. Blue lines are those who passed through the filter. Gray circles are all the real features, blue squares are those ones that were detected. Red triangle is the laser rangefinder location.

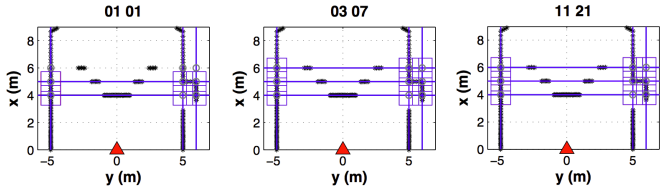

Fig. 7. A scene with detected features. Black points are raw measurements. Gray lines are those that was filtered. Blue lines are those who passed through the filter. Gray circles are all the real features, blue squares are those ones that were detected. Red triangle is the laser rangefinder location.

Therefore the analysis was based on variation in the resolution of the Hough Transform (Δd y $\Delta\theta$) and variation in the neighborhood of peaks (ΔV_d and ΔV_θ). The other parameters were set experimentally in the following values:

- $MinTh = 4$.
- $MaxN = 8$.
- $MinDist = 0.04m$.
- $MinPuntos = 4$.
- $MaxError = 0.02m$.
- $MinAng = 15^o$

IV. RESULTS

A. Simulated Testbench

In order to perform a graphical analysis several laser scans were obtained with the simulation software. Below is an analysis of a couple of them.

The Fig. 6 shows a single scan analyzed with different parameters. The left one has the smallest neighborhood ($\Delta V_d = 1$ and $\Delta V_\theta = 1$), and the right one the greater ($\Delta V_d = 11$ and $\Delta V_\theta = 21$).

In all three cases, gray lines are eliminated by the filter since they are "bad copies" of the actual line (right vertical). In addition, the larger the neighborhood less false (gray) lines appear. The more false lines extracts the Hough transform, the least amount of actual lines are detected. So we have to fine tuning the values for the peaks neighborhood.

The Fig. 7 shows another single scan.

To compare numerically the results were used parameters similar to those proposed in [11] ,[3] adapted to the extraction of point references as in [12]. The algorithms were compared for the amount of extracted features, the correctness and the execution time. The determination of the parameters of comparison was made using a test bench of more than 250 scans obtained by 2D simulation. Table I shows the results obtained for each implementation.

TABLE I
RESULTS OF SIMULATED TESTBENCH

Neigh.			Lines		features		correctness	
V_θ	V_d	Time [ms]	Detected	Filtered	Matched	Extracted	T.P. [%]	F.P. [%]
Resolution: $\Delta\theta = 1^o$ y $\Delta d = 1cm$								
1	1	151	2000	1090	496	1000	16,8	50,4
11	21	144	1871	969	915	1019	31,0	10,2
33	67	140	1721	1109	1147	1376	38,9	16,6
Resolution: $\Delta\theta = 2^o$ y $\Delta d = 2cm$								
1	1	80	2000	979	954	1116	32,3	14,5
11	21	80	1844	1209	1426	1481	48,3	3,7
33	67	78	1668	1237	1491	1565	50,5	4,7
Resolution: $\Delta\theta = 3^o$ y $\Delta d = 2cm$								
1	1	74	1995	1103	1134	1381	38,4	17,9
5	7	72	1753	1207	1498	1511	50,8	0,9
15	21	70	1582	1237	1566	1577	**53,1**	**0,7**
Resolution: $\Delta\theta = 5^o$ y $\Delta d = 2cm$								
1	1	67	1850	1298	1402	1760	47,5	20,3
3	7	65	1573	1232	**1577**	1577	**53,5**	**0,0**
11	21	64	1464	1244	**1601**	1601	**54,3**	**0,0**
Resolution: $\Delta\theta = 5^o$ y $\Delta d = 5cm$								
1	1	64	1907	1102	1328	1328	45,0	0,0
3	5	63	1905	1220	1480	1485	50,2	0,3
11	11	62	1797	1338	**1646**	1763	55,8	6,6
Resolution: $\Delta\theta = 5^o$ y $\Delta d = 10cm$								
1	1	62	2000	845	838	841	28,4	0,4
3	3	62	2000	948	904	955	30,6	5,3
11	7	62	1980	1190	1128	1542	38,2	26,9
Real lines: 1700					Real features: 2950			

The execution time measurement was performed using the functions *tic()* and *toc()* in Matlab. The time shown is obtained by dividing the total execution time among the number of scans stored in the test bench. The correctness measures are defined as follows [11]:

$$TrueP = \frac{NumMatches}{NumTrueFea}\% \quad (3)$$

$$FalseP = \frac{NumFeaExByAlgo - NumMatches}{NumFeaExByAlgo}\% \quad (4)$$

where:

- $NumTrueFea$: is the number of real features.
- $NumFeaExByAlgo$: is the number of features extracted by an algorithm.
- $NumMatches$: is the number of matches to real features

As seen in Table I, for greater neighborhood, the true positives rate is greater and the false positives rate is less. logically, the processing time is greater for better resolutions and for poor resolutions, the results are deficient.

B. Real Testbench

In order to validate the feature extraction method developed. It was tested with more than 40 real scans. Results are shown in Table II. In this case, we include a new measure: the number of blind scans. These are scans where the method

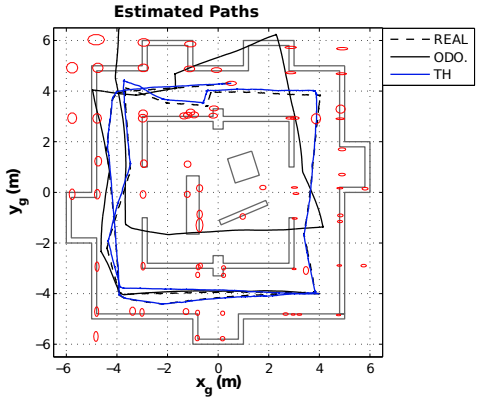

Fig. 8. The map of the environment appears in watermarks. Real path (black dashed line). Odometry estimated path (black line). EKF-SLAM estimated path (blue line). Extracted features (red ellipses).

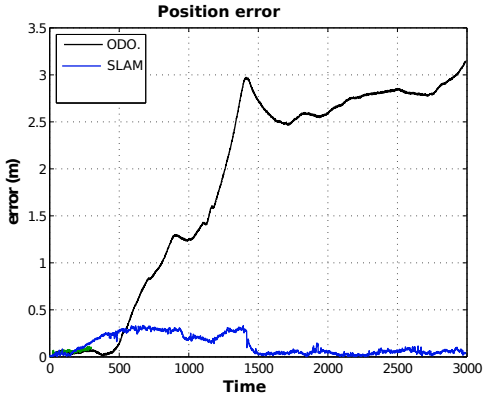

Fig. 9. Estimated position error for odometry estimated path (black) and slam estimated path (blue).

was unable to find any feature. Better results are obtained using the best resolution and the greater neighborhood.

There are a great quantity of blind scans. It is caused by the large hallway where only parallels lines are detected (blue zone in Fig. 5).

TABLE II

RESULTS OF REAL TESTBENCH

Parameters				blind scans	Features			correctness	
					Real	Matches	Extracted	T.P.	F.P.
$\Delta\theta$	Δd	V_θ	V_d					[%]	[%]
0,5	1	61	101	11	163	135	157	82,8	14,0
0,5	1	61	51	11	163	135	168	82,8	19,6
2	5	31	51	10	163	126	152	77,3	17,1
3	2	15	21	10	163	122	172	74,8	29,1
5	2	3	7	15	163	091	126	55,8	27,8
5	2	11	21	11	163	121	170	74,2	28,8

C. Simulated SLAM

A standard EKF-SLAM algorithm was implemented, similar to those shown in [1][13]. The features used as observations to the EKF were extracted with the proposed method. In Fig. 8 is shown the simulated environment. As observed, the path estimated by pure odometry, diverge largely to the real path. Instead, the EKF-SLAM estimated path converges to the real path. Further in the Fig. 9 shows the evolution over time of the estimated position error.

V. CONCLUSIONS AND FUTURE WORK

The method developed in this work proved to be fully functional to be used as part of complex navigation algorithms. Here was tested with EKF-SLAM but can be used in any feature-based algorithm.

It has a good rate of true detections and it is robust to the uncertainty of the sensor. However it has 2 disadvantages. First, it has a high computational cost and second it is weak in situations dominated by parallel lines (no intersection points). The latter disadvantage is almost nonexistent when using a large ROV laser rangefinder.

In the future, we will implement SLAM in real environments and we will compare the performance of the method based on Hough Transform with other feature extraction methods

ACKNOWLEDGMENT

Authors want to thank to the Research and Development Deanship (DID) of Simon Bolivar University for the financial support to this project.

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localisation and mapping (SLAM): Part I the essential algorithms," *Robotics and Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.128.4195

[2] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108–117, Sep. 2006. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1678144

[3] P. Núñez, R. Vázquez-Martín, J. C. Toro, A. Bandera, and F. Sandoval, "Natural landmark extraction for mobile robot navigation based on an adaptive curvature estimation," *Robotics and Autonomous Systems*, vol. 56, pp. 247–264, 2008.

[4] N. Werneck and A. Costa, "Monocular visual mapping with the fast hough transform," in *VI Workshop de Visão Computacional*, Jul 2010, pp. 279–284. [Online]. Available: http://iris.sel.eesc.usp.br/wvc/anais_WVC2010/artigos/oral/72827.pdf

[5] A. Santana and A. Medeiros, "Monocular-slam using floor lines," in *Robotics Symposium and Intelligent Robotic Meeting (LARS), 2010 Latin American*, oct. 2010, pp. 97 –102.

[6] J. Pérez Lorenzo, R. Vazquez-Martin, P. Nunez, E. Perez, and F. Sandoval, "A hough-based method for concurrent mapping and localization in indoor environments," in *Robotics, Automation and Mechatronics, 2004 IEEE Conference on*, vol. 2, dec. 2004, pp. 840 – 845.

[7] D. Ribas, P. Ridao, J. Neira, and J. D. Tardós, "A Method for Extracting Lines and their Uncertainty from Acoustic Underwater Images for SLAM," in *6th IFAC Symposium on Intelligent Autonomous Vehicles (IAV'07)*, Toulouse, France, September 2007.

[8] Y. Li and E. B. Olson, "Extracting general-purpose features from LIDAR data," in *IEEE International Conference on Robotics and Automation*, 2010, pp. 1388 – 1393. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5509690

[9] C. Tomasi and T. Kanade, "Detection and tracking of point features," International Journal of Computer Vision, Tech. Rep., 1991.

[10] R. Duda and P. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972. [Online]. Available: http://dl.acm.org/citation.cfm?id=361242

[11] 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*, aug. 2005, pp. 1929 – 1934.

[12] N. Certad, R. Acuna, A. Terrones, D. Ralev, J. Cappelletto, and J. Grieco, "Study and improvements in landmarks extraction in 2d range images based on an adaptive curvature estimation," in *VI Andean Region International Conference ANDESCON 2012*, Cuenca, Ecuador, nov 2012, pp. 95–98.

[13] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms and Implementation*, H. Choset, Ed. The MIT Press, 2005.

Novel Certad was born in Caracas, Venezuela in 1985. He obtained his Bachelors Degree in Electronics Engineering, and a M.S. Degree in 2013, from Simon Bolivar University, Venezuela in 2007 and 2013 respectively. He is currently working as Assistant Professor in the Electronics and Circuits Department, in the Simon Bolivar University. His research interest includes: mobile robots, SLAM, autonomous navigation and sensor fusion.