VI CONGRESSO NACIONAL DE ENGENHARIA MECÂNICA
VI NATIONAL CONGRESS OF MECHANICAL ENGINEERING
18 a 21 de agosto de 2010 – Campina Grande – Paraíba - Brasil
August 18 – 21, 2010 – Campina Grande – Paraíba – Brazil

A MODIFIED INCREMENTAL ALGORITHM FOR LINE-BASED MAP USING SONAR RANGE DATA

Claudia Patricia Ochoa Diaz, claudiaochoa@unb.br[1]
Alberto José Álvares, alvares@alvarestech.com[1]
[1] Departamento de Engenharia Mecânica e Mecatrônica, Universidade de Brasília (UnB), Campus Darcy Ribeiro - CEP 70910-900

Abstract. *This paper presents a line-extract algorithm based on the classic incremental algorithm. The proposed modification enhance the performance of the algorithm in terms of sensitivity to the parameters, and inclusion of outliers. A subsequent merging process is performed in order to cluster those extracted lines that match to the same line, in this way the volume of extracted features is reduced without affecting the resemblance of the constructed map with the real environment. This algorithm is part of a mapping strategy for a 2D feature-based map using ultrasonic sensor measurements. Simulation results show that the proposed modification has a good performance compares to the classic approach.*

Keywords: *line extraction, incremental algorithm, sonar data*

1. INTRODUCTION

A map is the result of collecting measurements from a certain sensory system, put them in an appropriate form and extracted the important information that results in a coherent representation of the environment explored by the robot. There are two different approaches for representing environments: *occupancy grid* maps and *feature-based* maps. The *grid-based* maps build a discrete representation using two or three-dimensional grid of cells. Each cell stores an occupation state of the space represented by that cell, which can be seen as the probability of being occupied or free space. The success of this kind of technique depends mainly on the cell size.

The *feature-based* map represents different objects of the environment as a set of *features*, which are recognizable structures of elements that can be extracted from raw data and are mathematically described (Siegwart and Nourbakhsh (2004)). The detection of these elements, also called *geometric primitives*, depends on the type of sensor used and the implemented extraction algorithm. The most common features are points, lines, arcs and some simple polygons, which correspond to corners, walls, edges and other real structures. This type of map needs less computational resources than the occupancy grid map due the reduction of the raw data, which also reduce the presence of spurious data entailed by environments conditions or by the sensor system itself.

Among many geometric primitives, line segment is the simplest one. All indoor environments has at least one structure which can be represented in a 2D map as line segment. Nguyen *et al.* (2005) present an experimental evaluation of the 6 most used line extraction algorithms. The drawbacks and the advantages of each approach are discussed by means of the computation of its speed, complexity, correctness and precision.

Many works have been proposed using detection of line segments for mapping purposes. The early work of Crowley (1989) models an indoor environment with extracted line segments from adjacent co-linear range measurements and matches these lines to a local model of the environment using a Kalman filter. Leonard and Durrant-Whyte (1992) introduce the concept of regions of constant depth (RCD) from the observations of sonar data, and from these regions, lines and point features were extracted. A more recent work is developed by Borges and Aldon (2004), presenting a feature detection framework using a 2D laser range finder. A Split-and-Merge Fuzzy (SMF) line extractor is proposed and compared it with another two classic kernels for line detection.

The Incremental algorithm, also known as Line-Tracking (Siadat *et al.* (1996)) has been mostly used as a segmentation algorithm for map building. The main advantage of this approach is its simplicity in terms of implementation, as well as good precision. The results presented in Nguyen *et al.* (2005) show that this type algorithm is also very fast and presents a low number of false positive. However, a critical drawback of the classic incremental algorithm is its sensitive of the established threshold for data evaluation.

On the other hand, the ultrasonic sensor, also known as *sonar* (stands for Sound Navigation And Ranging), has been used in mobile robotics mostly as an avoidance-obstacle sensor, while for mapping purposes, the use of this kind of sensor has been left behind. Despite of its limitations, (these are, specular reflections and no direct angular information) the sonar provide a direct depth information about the structures presented in the environment and, with the appropriated representation of its measurements, it can be used for constructing an accurate map, resulting in a low-cost solution. Some interesting works using ultrasonic sensor are presented by Barshan and Kuc (1990),Lee *et al.* (2005) and Wang *et al.* (2007), which reaffirm the use of this sensor for mapping purposes.

In this paper, a strategy for extracting line features based on a modified incremental algorithm is presented. The paper is organized as follows: the sensor and the environment model are discussed in section 2. Section 3 presents the incremental algorithm and the proposed modification. Simulation results and their discussion appears in section 4, and finally, in section 5 the conclusions are drawn.

2. MEASUREMENT AND ENVIRONMENT REPRESENTATION

2.1 Sonar Range Model

The ultrasonic sensor is a time-of-flight active ranging sensor that uses the speed of propagation of the sound to provide the distance information of detected objects. The sound waves propagate in a cone-like pattern with opening angles between 20 to 40 degrees (Nomadic (1999)).

A typical measurement profile of an ultrasonic sensor is shown in Fig.1a, the observation is constituted by the measured distance d between the detected obstacle and the sensor, also known as the *range reading*. It's important to point out that the sonar is not capable of provide the orientation of the detected object ϕ, instead of this, this value can be estimated as being the sonar orientation φ, with an uncertainty σ_φ given by the cone model limited by the angular beam β.

Therefore, it's possible to express the n^{th} sonar measurement as

$$d_n = D_n + \varepsilon_d, \quad \varepsilon_d \sim N(0, \sigma_d^2) \tag{1}$$

where D_n corresponds to the real distance value which is affected by a measurement error ε_d, assumed to have a normal distribution with zero-mean and variance given by the distance uncertainty σ_d^2. To facilitate the task of line extraction, all the measurements should be in the same reference system in order to compare all of them and identify those who represent collinear points. Therefore, observations are projected into a common reference, that is, the global coordinates system. Considering the n^{th} measurement as a point data (see Fig. 1b), its projection on the cartesian global frame corresponds to

$$\{p_n\}^G = \begin{bmatrix} x_n \\ y_n \end{bmatrix}^G = \begin{bmatrix} (r + d_n)\cos(\phi_i + \theta_R) + x_R \\ (r + d_n)\sin(\phi_i + \theta_R) + y_R \end{bmatrix} \tag{2}$$

The measurement error is also propagated from the sensor frame to the global coordinates system according to Navarro *et al.* (2008),

$$\{C_{xy}\}^G = \frac{\partial \{p_n\}^G}{\partial \{x_n\}^R \partial \{y_n\}^R} \{C_{xy}\}^R \left[\frac{\partial \{p_n\}^G}{\partial \{x_n\}^R \partial \{y_n\}^R}\right]^T + \frac{\partial \{p_n\}^G}{\partial x_R \partial y_R \partial \theta_R} \{C_{Robot}\} \left[\frac{\partial \{p_n\}^G}{\partial x_R \partial y_R \partial \theta_R}\right]^T \tag{3}$$

The first term of Eq. (3) corresponds to the propagation of the covariance error matrix in the robot frame $\{C_{xy}\}^R$, while the second one is the propagation of the robot's covariance error $\{C_{Robot}\}$, which depends on the odometry system of the platform. With these results it's possible to represent any data point in the global reference along with its covariance error matrix.

2.2 Line Segment Model

As it was mentioned before the representation of the environment is by means of a set of line segments. For the line representation, the normal form has been chosen. This form can represent all kinds of line segments, including the lines parallel to the y axis. The expression for the line model is

$$\rho = x_n \cos(\alpha) + y_n \sin(\alpha) \tag{4}$$

where ρ and α constitute the model parameters, this small number let its estimation process perform very fast. For this purpose, a robust estimation method is used, specifically, the *M-Estimators*. This technique for parameter estimation is defined as the solution that minimized (Huber (1981)):

$$J = \sum_{n=1}^{N} d(r_n) \tag{5}$$

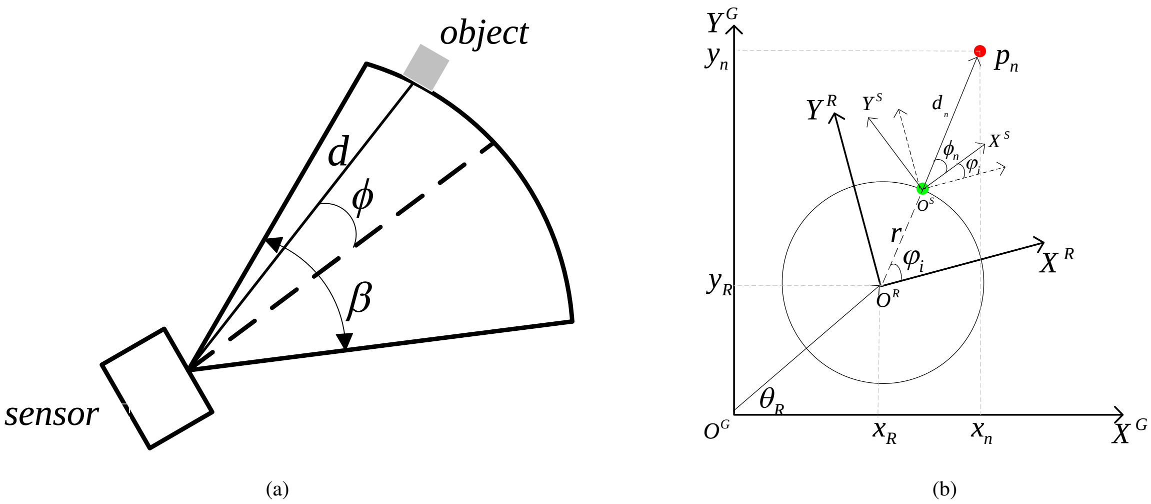


Figure 1: (a). Sonar Measurement Profile (b). Projection of to the global reference

where $d(r_n)$ is a cost function associated to the residuals or distances between the observation $\{p_n|\, n = 1, ..., N\}$ and its model. This function tries to reduce the effects caused by the presence of outliers in p_n. The minimization of Eq.(5) is solved by finding $\mathbf{\Theta}$, the parameters vector, such that

$$\sum_{n=1}^{N} \frac{\partial d(r_n)}{\partial r_n} \frac{\partial r_n}{\partial \mathbf{\Theta}} = \sum_{n=1}^{N} \psi(r_n) \frac{\partial r_n}{\partial \mathbf{\Theta}} = 0 \tag{6}$$

where $\psi(r_n) = \partial d(r_n)/\partial r_n$ is called the *influence function*. This function can also be defined by a weight function w_n, like this

$$\psi\,(r_n) = w(r_n) r_n \tag{7}$$

Substituting Eq. (7) in Eq. (6), we have

$$\sum_{n=1}^{N} w(r_n) r_n \frac{\partial r_n}{\partial \mathbf{\Theta}} = 0 \tag{8}$$

which lead us to solve an *iterative reweight least squares* (IRLS) problem (L.Ljung (1999)). The following equation is the cost function used for the line fitting problem:

$$d(r_n) = \hat{\rho} - x_n \cos(\hat{\alpha}) - y_n \sin(\hat{\alpha}) \tag{9}$$

$\hat{\rho}$ and $\hat{\alpha}$ correspond to the estimated parameters that has been calculated in the current iteration, whose computation follows the expressions given by

$$\begin{aligned} \hat{\rho} &= \bar{x}\cos(\hat{\alpha}) - \bar{y}\sin(\hat{\alpha}) \\ \hat{\alpha} &= \frac{1}{2}\arctan\left(\frac{-2\sum_n w_n (x_n - \bar{x})(y_n - \bar{y})}{\sum_n w_n (y_n - \bar{y})^2 - \sum_n w_n (x_n - \bar{x})^2} \right) \end{aligned} \tag{10}$$

where $\bar{x}$ and $\bar{y}$ are the weighted mean of the x-coordinate and y-coordinate, respectively.

3. INCREMENTAL-BASED ALGORITHM FOR LINE EXTRACTION

3.1 The Classic Incremental Algorithm

The incremental algorithm is sequential, this means its inputs are processed one by one according to the order in which were taken. In this way, there's no need to wait until the whole data set is colletced. This behavior results very useful for *on-line* applications. Figure (2) shows an ilustrative situation of this.

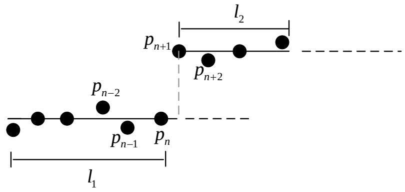


Figure 2: Illustrative situation of the Incremental algorithm

The basic idea behind the incremental algorithm is starting from the first two points of the data set, and assuming that both are collinear points, a line is constructed. A next point is added and is accepted as a support point of the estimated line if the distance between the point and the line don't exceed an established threshold D_{max}. If the current point is accepted, the parameters of the line are recomputed and the next point is analyzed. If not, the current point is taken as being part of a new line. Figure (3) shows the implementation of the algorithm.

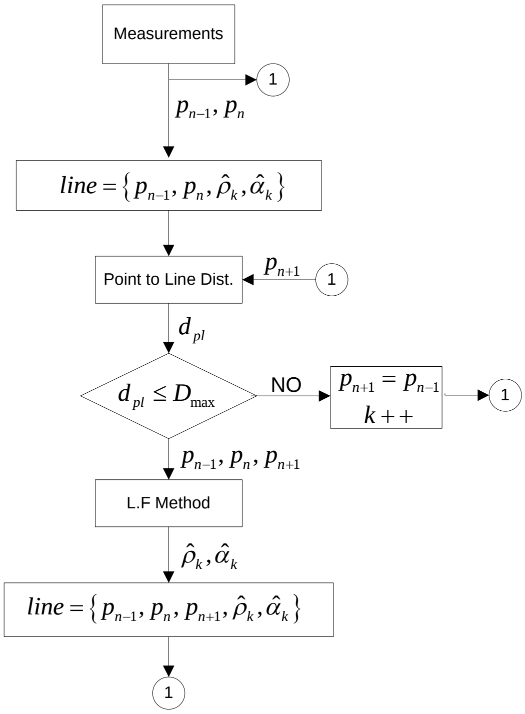


Figure 3: Classic Incremental Algorithm

The above algorithm is repeated until the last point of the data set is analyzed. The LINE-POINT DISTANCE procedure correspond to the calculation of the residuals in Eq. (9), while the L.F METHOD or Line Fitting Method, is the estimation of the line parameters (see Eq. (10)).

3.2 The Proposed Algorithm

As it was mentioned before, the main drawback of the incremental algorithm is the adjustment of the distance threshold, which leads to segment a single line into different lines or merged different lines into one, besides of the sensitivity to spurious data. To overcome those problems, our approach makes a modification in the basic algorithm and introduce an extra condition to evaluate the inclusion of new supports points in some line. First, no assumption in the initial points is used (in the classic approach the first two points are assumed to belong to the same line), instead of this, an euclidean distance between consecutive points is calculated,

$$d_{euc} = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \tag{11}$$

This distance is used to evaluate the first two points of some new line, if d_{euc} doesn't exceed the threshold D_{max_1} these two points can be used to estimate a new line. If D_{max_1} is exceeded, then the first point is rejected and another point

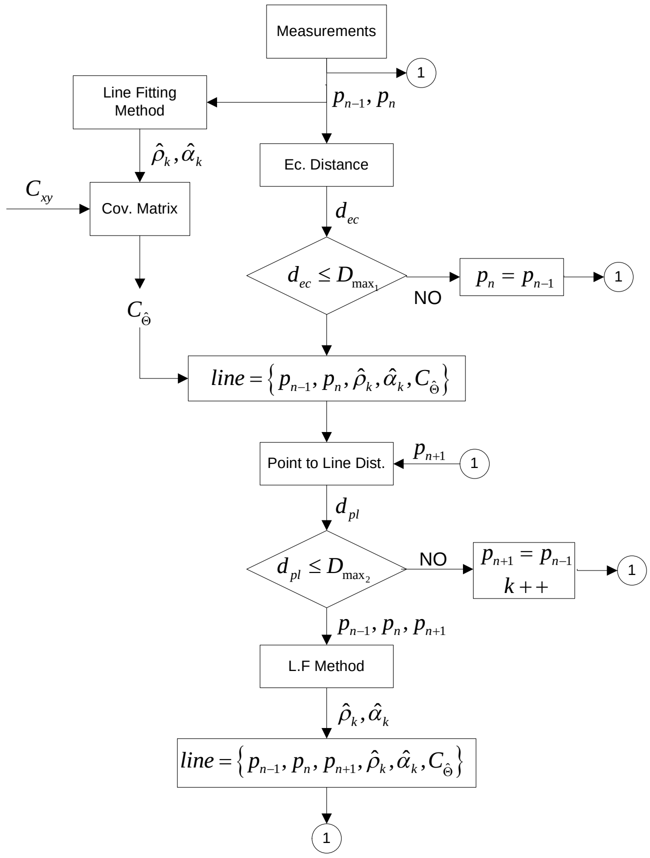


Figure 4: Modified Incremental Algorithm

is analyzed along with the not-rejected point. Once a new line is estimated, the point-to-line distance (the one used in the original version) is calculated in order to include (or not) new support points.

This might be seen as another parameter to be adjust, but D_{max_1} is very simple to calculate because it's related to the measuring process. To be specific, this distance threshold could be calculated from the time elapsed between two consecutive measurements, known as *timestamp*, and the maximal value for the translational velocity of the robot. The value of D_{max_1} will remain fixed as long as the velocity of the robot is constant. The modification is shown in figure (4)

The inputs of the segmentation algorithm are the measurements in cartesian coordinates, the threshold for the euclidean distance evaluation between consecutive points D_{max_1}, the threshold for the line-to-point distance D_{max_2} (the same of the classic approach), and the covariance matrix of the measurements, calculated in (6). This last input is necessary in order to perform the COV_MATRIX procedure, which is the computation of the covariance matrix of the estimated parameters. This matrix is used to establish the limits for the value of D_{max_2} and is calculated using the Haralick's methods (Haralick (1994)). This method propagates the uncertainty of $p_n = \{x_n, y_n\}$, represented in C_{xy}, to the estimated parameters of the line which minimize an implicit cost function. As a result of this stage, different lines are detected from the whole data set along with their covariance matrix $C_{\hat{\mathbf{\Theta}}_k}$. Figure 4 presents a representation of the proposed modification.

3.3 Merging Process

After all the lines are extracted from the raw data, it is necessary to identify the ones that belong to the same feature, group them (according to a certain criterion) and estimate a single line that best represents the group. As the measurements are in the same frame, that is the global reference, the task of matching the initial lines is facilitated in someway. The criterion of decision to merged lines is a chi-squared test. The criterion proposed by Pfister *et al.* (2003) is used. Here, the test determine if the difference between two lines is within the 3σ deviance threshold defined by the combined uncertainties of the lines:

$$\chi^2 = \left(\delta\hat{\mathbf{\Theta}}\right)^T \left(C_{\hat{\mathbf{\Theta}}_k} + C_{\hat{\mathbf{\Theta}}_{k+1}}\right)^{-1} \left(\delta\hat{\mathbf{\Theta}}\right) < 3 \tag{12}$$

where

$$\delta\hat{\mathbf{\Theta}} = \begin{bmatrix} \hat{\rho}_{k+1} - \hat{\rho}_k \\ \hat{\alpha}_{k+1} - \hat{\alpha}_k \end{bmatrix} \tag{13}$$

If the condition is satisfied the lines are sufficiently similar to be merged.

4. RESULTS

4.1 Simulated Environment

To validate the proposed strategy, a simulated environment was created. In this simulation our robot platform travels among the whole environment composed by 13 walls and a space between two of them that represents an open door. The simulated environment is shown in figure (5a). For the perception task, 8 sonars are used in the same configuration the real robot has. The robot pose is measured in intervals of $50\mathrm{mm}$, which is the traveled distance between consecutive scans.

To give more realism to the data set, a measurement error is added according to Eq.(1). Each sensor has a maximum distance range of $6.5\mathrm{m}$. The uncertainty in the distance measure is set to $40\mathrm{mm}$, which correspond to the value extracted from a calibration test. Figure (5b) illustrates the collected dataset, which is the result of 63 scans of the whole environment, giving a total of 504 observations. It can be observed that not all the lines of the environment are seen by our robot (a very normal situation), therefore, only 8 of the 13 lines can be detected from the raw data, these are line 1, 4, 5, 8, 10, 11, 12 and line 13.

4.2 Results and Discussion

The preprocessing of data and the implementation of the line-extraction algorithm is carried out in Matlab. The algorithm parameters are chosen according to the sensor hardware, the configuration of the simulation and the environmental conditions. D_{max_1} was adjusted to the same value of the traveled distance between scans ($50\mathrm{mm}$), while D_{max_2} depends on the uncertainties of the parameters of the estimated line ($\sigma^2_{\hat{\rho}}$ and $\sigma^2_{\hat{\alpha}}$), extracted in the incremental algorithm. It's also important to set a minimum number of support points per line in order to disregard those lines that have a few number of data, which can represent erroneous extracted features. This parameter was set to 5.

To compare the performance of the classical algorithm with the proposed modification, the precision of both algorithms is calculated. Therefore, two error were defined in the same way as Nguyen *et al.* (2005):

$$\begin{aligned} \Delta\rho_k &= \rho_{real,k} - \hat{\rho}_m, \quad m = 1, ...M \\ \Delta\alpha_k &= \alpha_{real,k} - \hat{\alpha}_m, \quad m = 1, ...M \end{aligned} \tag{14}$$

where m is the number of matched pair, $\rho_{real,k}$, $\alpha_{real,k}$ are the true line parameters, $\hat{\rho}_m$ and $\hat{\alpha}_m$ are the parameters of the corresponding matched line. The precision is finally calculated by means of the computation of the variances of both parameters:

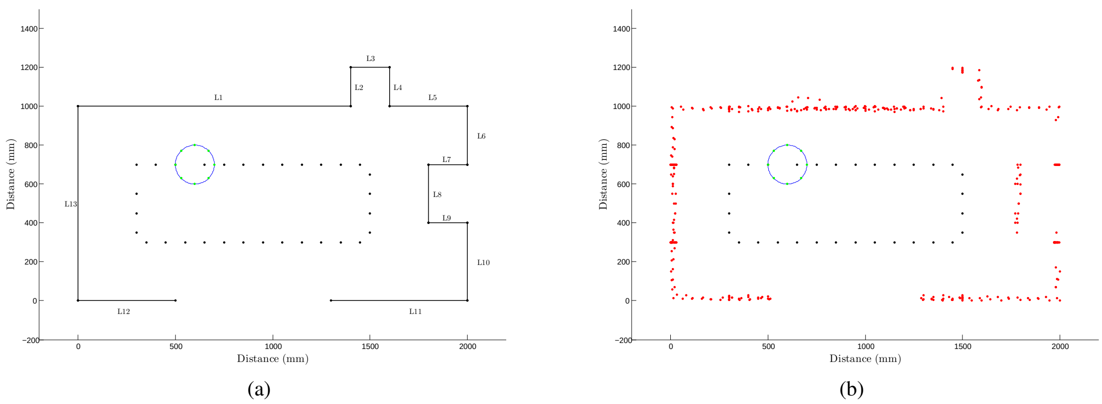
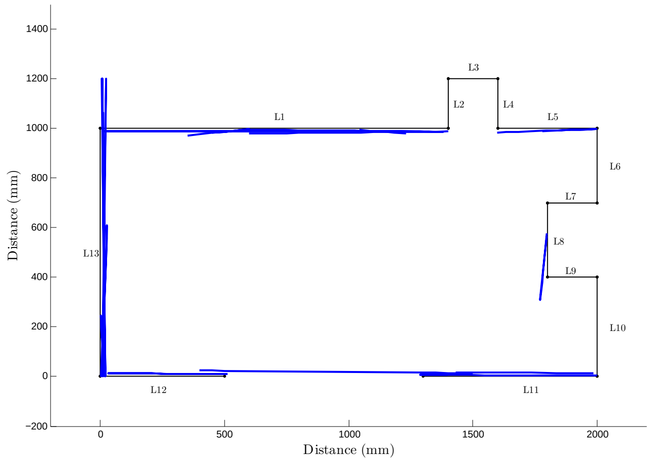


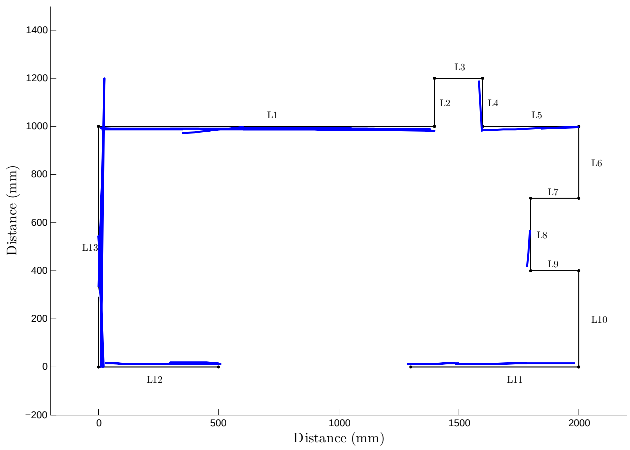
Figure 5: (a). Simulation Environment (b). Sensors Measurements

$$\begin{aligned} \bar{\Delta}\rho_k &= \frac{1}{M}\sum(\Delta\rho_k)_m \quad \sigma^2_{\Delta\rho} = \frac{1}{M-1}\sum\left((\Delta\rho_k)_m - \bar{\Delta}\rho_k\right)^2 \\ \bar{\Delta}\alpha_k &= \frac{1}{M}\sum(\Delta\alpha_k)_m \quad \sigma^2_{\Delta\alpha} = \frac{1}{M-1}\sum\left((\Delta\alpha_k)_m - \bar{\Delta}\alpha_k\right)^2 \end{aligned} \tag{15}$$

Figure (6) presents the results of both algorithms, the classic one and the proposed modification. Moreover, Table 1 presents the experimental results and the performance of the two algorithms by means of the precisions of the estimates lines. It can be observed that the classic approach was not able to detect all the line features seen by the robot, specifically the lines 4 and 10. This is a typical consequence of the sensitivity to the parameter D_{max}. In this case, the value of D_{max} might be too conservative to admit the data points of lines 4 and 10 as being the support points of such lines. The other consequence of the value of D_{max} is reflected in line 8. It can be seen that the estimated line presents a notorious error in its orientation, which can be explained by the fact that some outliers were added to this line, affecting the line estimation. Moreover, because of the absence of an extra condition that evaluates the distance between consecutive points, the lines 11 and 12 were merged into one. All the situations mentioned before were remedied by the modifications made in the original algorithm, which is reflected in the result shown in figure (6b). Comparing the precision of both algorithms, the proposed algorithm presents better results, reflecting the advantages of the implemented modifications.



(a)



(b)

Figure 6: (a). Results from the classic Incremental Algorithm (b). Results from the proposed modification

5. CONCLUSION

This paper presents a Line Extraction algorithm based on the classic Incremental algorithm using sonar range data. The proposed approach exploits aspects of this algorithm like simplicity in implementation, speed and precision. However, the Incremental approach presents some drawbacks that affect its performance in some way,these are: susceptibility to the parameter and inclusion of outliers.

Table 1: Performance Results

Line	Real		Classic Approach				Proposed Modification			
	ρ_k	α_k	$\hat{\rho}_k$ (mm)	$\hat{\alpha}_k$ (rad)	$\sigma_{\Delta\rho}$	$\sigma_{\Delta\alpha}$	$\hat{\rho}_k$ (mm)	$\hat{\alpha}_k$ (rad)	$\sigma_{\Delta\rho}$	$\sigma_{\Delta\alpha}$
1	1000	1.571	987.731	1.575	133.106	0.179	987.1	1.576	93.565	0.158
4	1600	0	-	-	-	-	1651.4	0.057	0	0
5	1000	1.571	933.56	1.602	0	0	933.27	1.602	0	0
8	1800	0	1734	0.096	0	0	1781.1	0.0801	64.516	0.0434
11	0	1.571	-	-	-	-	6.662	1.558	10.284	0.012
12	0	1.571	-	-	-	-	7.2867	1.565	13.128	0.003
13	0	0	10.606	0.023	15.044	0.02	8.183	0.037	14.047	0.131

This work introduced an extra condition for data evaluation in order to reduce these effects. Also, no initial assumptions are made, avoiding the inclusion of outliers into the estimated line. Results shown that the proposed modifications indeed reduced all the aspects mentioned before, enhancing the performance of the original version of the Incremental algorithm. Moreover, a merging process is also implemented in order to cluster all the lines detected by the different sonars mounted in our robotic platform. In this way the number of lines detected is considerable reduced, resembling much more the built map with the real environment.

This work is part of a project that includes a map-based localization strategy and navigation in indoor environments. Future work is focused on the validation of the line-extraction algorithm on the robotic platform with real environments, and also the implementation of a localization approach based on the information provided by the constructed map.

6. REFERENCES

Barshan, B. and Kuc, R., 1990. "Differentiating sonar reflections from corners and planes by employing an intelligent sensor". *IEEE Transactions on Pattern Analysis and Machine Intellegence*, Vol. 12, No. 6, pp. 560–569.

Borges, G.A. and Aldon, M.J., 2004. "Line extraction in 2d range images for mobile robotics". *Journal of Intelligent and Robotic Systems*, Vol. 40, pp. 267–297.

Crowley, J., 1989. "World modeling and position estimation for a mobile robot using ultrasonic ranging". *IEEE Conference on Robotics and Automation, ICRA 89*, Vol. 3, pp. 1574–1579.

Haralick, R.M., 1994. "Propagating covariance in computer vision". In *Proceedings of 12th ICPR*. pp. 493–498.

Huber, P., 1981. "Robust statistics".

Lee, S.J., Cho, D.W., Chung, W.K., Lim, J.H. and Kang, C.U., 2005. "Feature based map building using sparse sonar data". In *IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 492–496.

Leonard, J.J. and Durrant-Whyte, H.F., 1992. "Direct sonar sensing for mobile robot navigation". Technical report, Kluwer Academic Publishers.

L.Ljung, 1999. "System identification".

Navarro, D., Benet, G. and Blanes, F., 2008. "Line-based incremental map building using infrared sensor ring". *IEEE International Conference on Emerging Technologies and Factory Automation. ETFA 2008.*, pp. 833–838.

Nguyen, V., Martinelli, A., Tomatis, N. and Siegwart, R., 2005. "A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics". *Conference on Intelligent Robots and Systems, IROS 2005*.

Nomadic, 1999. *Nomad XR4000 Hardware Manual Release 1.0*.

Pfister, S.T., Roumeliotis, S.I. and Burdick, J.W., 2003. "Weighted line fitting algorithms for mobile robot map building and efficient data representation". *IEEE International Conference on Robotics and Automation. ICRA 2003*.

Siadat, A., Kaske, A., Klausmann, S., Dufaut, M. and Husson, R., 1996. "An optimized segmentation method for a 2d laser-scanner applied to mobile robot navigation". In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 96*.

Siegwart, R. and Nourbakhsh, I., 2004. "Introduction to auntonomous mobile robots". The MIT Press.

Wang, H.M., Hou, Z.G., Ma, J., Zhang, Y.C., Zhang, Y.Q. and Tan, M., 2007. "Sonar feature map building for a mobile robot". In *IEEE International Conference on Robotics and Automation*. pp. 4152–4157.

7. RESPONSIBILITY NOTICE