A Robust Approach to Build 2D Line Maps From Laser Scans

Carlos Lara and Leonardo Romero

Universidad Michoacana de San Nicolas de Hidalgo,
Morelia, Mich., 58000, Mexico
calara@lsc.fie.umich.mx, lromero@zeus.umich.mx

Abstract. Mobile robots typically need a map of the environment to perform their tasks. In indoor environments, a 2D geometric map is commonly represented by a set of lines and points. In this work we consider a mobile robot with a laser range finder and the goal is to find the best set of lines from the sequence of points given by a laser scan. We propose a probabilistic method to deal with noisy laser scans, where the noise is not properly modeled using a Gaussian Distribution. An experimental comparison with a very well known method (SMSM) using a mobile robot simulator and a real mobile robot, shows the robustness of the new method. The new method is also fast enough to be used in real time.

1 Introduction

A considerable number of researchers had been using laser range finders in mobile robotics for indoor environments. Laser range finder measures the outline of objects with high resolution and accuracy. This information can be used directly to perform simple tasks. However in most of the mobile robot tasks is necessary to build a map of the environment.

Normally the laser range finder performs is in a plane parallel to the floor and each laser scan provides n points from the environment as it is shown in the figure 1(b). Usually points are expressed in polar format (r_i, α_i), where r_i is the distance from the sensor to the detected object at direction α_i (see figure 1(a)).

Because indoor environments have a lot of planar surfaces such as walls, doors, or cupboards; Line Maps (LM) are commonly used to represent them. LM are more compact and accurate than occupancy grids. Unfortunately, as state in [4], there are some difficulties to get the best LM from a laser scan: 1) find the best number of lines, 2) determine which points belong to which line, and 3) estimate the line model parameters given the points that belong to a line.

This paper introduces a new approach called WSAC to find the LM even in the presence of noisy measurements. The main idea is to perform a probabilistic search of a line Θ within a short sequence of consecutive measurements and then review if there are more points in the laser scan that could be represented by Θ. The final line is estimated using all the possible points.

Problems associated with outliers (atypical data) are handled by the probabilistic search and different segments of the same line are identified easily. Other

© *A. Gelbukh, S. Torres, I. López (Eds.)*
Advances in Computing Science and Engineering
Research in Computing Science 19, 2006, pp. 197-208

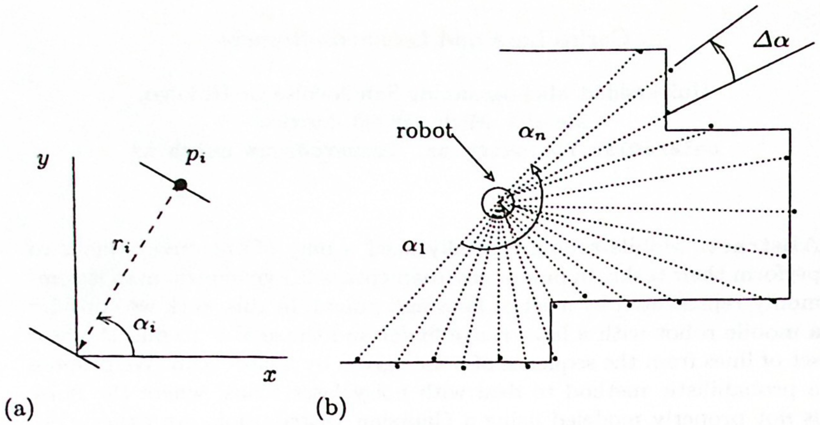


Fig. 1. Laser measurements

methods have difficulties with outliers or report many disconnected segments when in fact they belong to the same line.

The rest of this paper is organized as follows: section 2 reviews previous approaches to find a single line and multiple lines in a sequence of measurements. Section 3 describes the proposed method, Window SAmple Consensus WSAC. Experimental tests are reported in section 4 and some conclusions are given in section 5.

2 Previous Works

In this section we briefly review some previous methods to estimate a single line and multiple lines given a set of points.

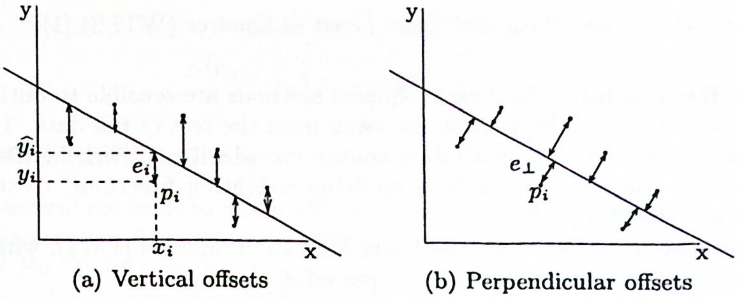
2.1 Fitting a Single Line

Least Squares Methods. The method of Least Squares (LS) proposed by Legendre and Gauss [14] assumes that the best-fit line is the line that minimizes the sum of squared deviations or errors from the line to each point in the set of points. We can state this formally:

$$\Theta = arg\min E(\Theta) = \sum_{i=1}^{n} e_i^2 \tag{1}$$

A common way to resolve eq. 1 is to calculate e_i as the vertical offset as is shown in the figure 2(a). In this case we have

$$e_i = y_i - (mx_i + b) \tag{2}$$



(a) Vertical offsets (b) Perpendicular offsets

Fig. 2. Different deviations from the line

where m and b are the parameters of the line given by $y = mx + b$. The solution of equation 1 using vertical offsets (eq. 2) can be found in several textbooks (v.g. [10]).

If we minimized perpendicular offsets $e_{\perp i}$, as indicated in the figure 2(b), instead of vertical offsets, the method is called Total Least of Squares (TLS). In this case the error is given by

$$e_{\perp i} = r - x_i \cos\alpha - y_i \sin\alpha \tag{3}$$

where, r and α are the parameters of the estimated line in polar form. Finally if instead of weighting all points equally, we use nonnegative constants or weights a_i associated with each point (x_i, y_i), equation 1 becomes

$$\arg\min_{\alpha,r} \sum_{i=1}^{n} a_i \left[r - x_i \cos\alpha - y_i \sin\alpha\right]^2 \tag{4}$$

and its solution is given by:

$$\alpha = \frac{1}{2}\arctan\left(\frac{-2\widetilde{S}_{xy}}{\widetilde{S}_{yy} - \widetilde{S}_{xx}}\right) \tag{5}$$

$$r = \widetilde{x}\cos(\alpha) + \widetilde{y}\sin(\alpha) \tag{6}$$

where

$$\widetilde{S}_{xx} = \sum_{i=1}^{n} a_i \left(x_i - \widetilde{x}\right)^2 \qquad \widetilde{x} = \frac{\sum_{i=1}^{n} a_i x_i}{\sum_{i=1}^{n} a_i}$$

$$\widetilde{S}_{yy} = \sum_{i=1}^{n} a_i \left(y_i - \widetilde{y}\right)^2 \qquad \widetilde{y} = \frac{\sum_{i=1}^{n} a_i y_i}{\sum_{i=1}^{n} a_i}$$

$$\widetilde{S}_{xy} = \sum_{i=1}^{n} a_i \left(x_i - \widetilde{x}\right)\left(y_i - \widetilde{y}\right)$$

This method is called Weighted Total Least of Squares (WTLS) [1].

Robust Regression. The Least–Squares methods are sensible to outliers [12]. An outlier is a single observation far away from the rest of the data. There are several alternatives to deal with data contaminated with outliers. M–Estimators [17, 6] reduce the effect of outliers applying weighting functions, reducing the problem to a WTLS problem.

Rousseeuw [12] introduced the Least Median of Squares (LMS), which minimizes the median of the sequence of squared deviations,

$$E = \mathbf{median}(e_1^2, e_2^2, \ldots, e_n^2) \tag{7}$$

The minimization of 7 can not be obtained analytically and therefore requires an iterative method.

Another alternative is the method RANdom SAmple Consensus (RANSAC) [3] described in the Algorithm 1. One of the strengths of RANSAC is its ability to estimate the parameters with accurancy even when outliers are present in the data set. Frequently RANSAC is a better choice than LMS in vision due it can be better adapted to complex data analysis situations [9].

Algorithm 1 RANSAC Algorithm

Input: $\mathcal{P} = \{(x_i, y_i)\,|i = 1 \ldots n\}$, and a number of tries m
Output: Line $\Theta^\bullet$

```
1. $j \leftarrow 0$, $n^\bullet \leftarrow 0$.
2. Repeat until $j < m$
   - $j \leftarrow j + 1$
   - Select randomly two points from $\mathcal{P}$ and compute the line parameters $\Theta_j$
   - Count the number of inliers $n_j$ given a user tolerance $t$
   - if $n_j > n^\bullet$ then $n^\bullet \leftarrow n_j$, $\Theta^\bullet \leftarrow \Theta_j$
3. Reestimate $\Theta^\bullet$.
```

Unfortunatelly, this approach has difficulties: if the threshold t is set too high then the model estimation can be very poor. n_j in algorithm 1 can be viewed as a cost function Θ given by

$$C_\Theta = \sum_{i=1}^{n} \rho(e_{\perp i}) \tag{8}$$

where

$$\rho(e_{\perp i}) = \begin{cases} 1 & \text{if } e_{\perp i} < t \\ 0 & \text{elsewhere} \end{cases} \tag{9}$$

and $e_{\perp i}$ is the orthogonal distance of the i–th point to the line Θ. However, rather than using eq. 9, Torr and Zisserman in [15] proposes a better method called MSAC which uses the following cost function:

$$\rho(e_{\perp i}) = \begin{cases} e_{\perp i} & \text{if } e_{\perp i} < t \\ t & \text{elsewhere} \end{cases} \qquad (10)$$

2.2 Finding Multiple Lines

There are several methods to find multiple lines [11, 13]. The most popular methods are: *Expectation–Maximization* (EM) [8], *Line Tracking* (LT) [2], *Iterative End Point Fit* (IEPF) [2], the Hough Transform (HT) [7, 5, 16] and the *Split–Merge Split–Merge* algorithm (SMSM) [18]. This section describes briefly the IEPF and SMSM algorithms due to its popularity and simplicity.

The method *Iterative End Point Fit* is illustrated in figure 3 and is explained in algorithm 2.

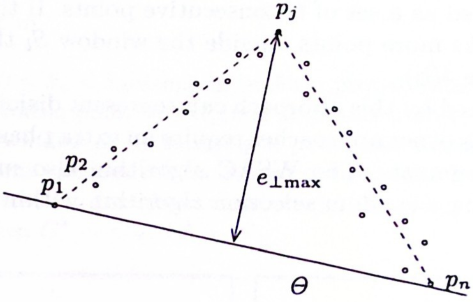


Fig. 3. *Iterative End Point Fit*(IEPF)

Algorithm 2 IEPF Algorithm

Input: The laser scan $S = \{(r_i, \alpha_i) | i = 1 \dots n\}$ and a threshold t
Output: The map of lines $\mathcal{M}$

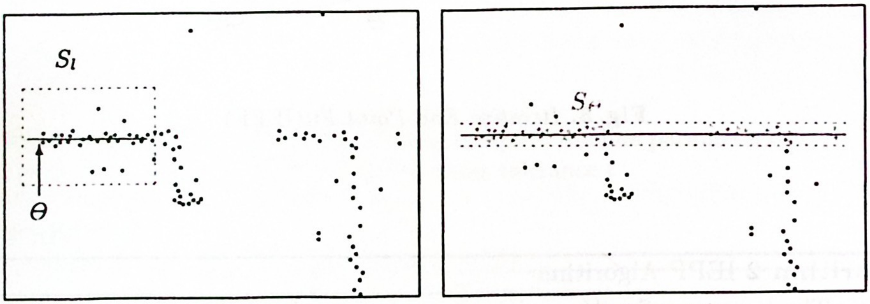
```
1. Initialize a list $\mathcal{L}$ with $\mathcal{P}$, $\mathcal{M} \leftarrow \{\}$
2. While $\mathcal{L} \neq \{\}$
   (a) Get the next set $S_j$ in $\mathcal{L}$
   (b) Find the line $\Theta$ which joins the first and the last point of $S_j$.
   (c) Detect the point $(r_k, \alpha_k)$ with maximum distance $e_{\perp max}$ to the line $\Theta$.
   (d) Remove $S_j$ from $\mathcal{L}$.
   (e) If $e_{\perp max} > t$ then
       - Add $S_{j0} = \{(r_j, \alpha_j) | j = 1 \dots k-1\}$ and $S_{j1} = \{(r_j, \alpha_j) | j = k \dots n_j\}$ to $\mathcal{L}$.
   (f) Else
       - Fit a line $\Theta_j$ to all points in $S_j$ (v.g using LS) and put $\Theta_j$ into $\mathcal{M}$.
3. Merge collinear segments in $\mathcal{M}$.
```

The *Split–Merge Split–Merge* algorithm (SMSM) [18] is an extended and more robust version of the IEPF algorithm. The first step consists in applying a *Breakpoint Detector* [1]. The idea behind this step is to detect and eliminate outliers, because they are not going to be included in any cluster. Then it merges two consecutive clusters if their distance (the distance between the final point of the first cluster and the first point of the second cluster) is less than a predefined threshold. In the second phase, SMSM applies the IEPF algorithm to all clusters. Finally it combines collinear segments.

3 A New Approach: WSAC

The main idea of the WSAC algorithm can be described focusing on how it extracts a single line. To extract a line the algorithm takes advantage of the order of the laser scan, searching a line Θ in a window S_l as shown in figure 4. A window S_l is defined as a set of t_l consecutive points. If the algorithm finds a line in S_l then it looks more points outside the window S_l that fit with the line Θ, as shown in figure 4(b).

The line Θ obtained by this approach can represent disjoint regions as shown in figure 4(b), whereas other approaches require an extra phase of post-processing to merge collinear segments. The WSAC algorithm also manages the outliers appropriately by using a random selection algorithm within window S_l.



(a) Local search (b) Global fit

Fig. 4. WSAC

Algorithm 3 explains the local search shown in the figure 4(a). It has a random algorithm similar to MSAC. To improve results it includes a mechanism of a ”sliding window". It means that the window can move slightly through the laser scan.

A local search is successful if the weighted consensus $C^\bullet$ of the line $\Theta^\bullet$ is greater than some threshold C_{min}.

If the local search was successful the algorithm performs the global fit. Otherwise it performs another local search in a different window.

The global search has three steps. Fist the algorithm determines the set of points S_Θ that support line $\Theta^\bullet$ by searching into the whole laser scan $\mathcal{P}$. Second, it removes the points which belong to small length segments from S_Θ. This step finds the segments of S_Θ by applying a BreakPoint Detector algorithm similar to the one presented in [1]. Finally the line parameters are recomputed using the set of inliers S_Θ and the line is added to the Map. An important consideration is that after the global search the points in S_Θ are removed from the laser scan $\mathcal{P}$.

Algorithm 3 WSAC: Searching a line within a sliding window

Input: The laser scan $\mathcal{P} = \{(x_i, y_i)|i = 1 \dots n\}$, the reference point l, a sliding constant k_d, a window size t_l, and a number of tries m
Output: A line $\Theta^\bullet$ with consensus $C^\bullet$

```
1. $j \leftarrow 0$, $C^\bullet \leftarrow 0$.
2. Repeat until $j < m$
   - $j \leftarrow j + 1$
   - Compute $k_s$, $0 \leq k_s \leq k_d$ using an uniform probabilistic distribution
   - Compute the initial point by skipping $k_s$ points from $l$
   - Compute the window $S_l$ by selecting $t_l$ points starting from the initial point
   - Select two points randomly from $S_l$ and compute the line parameters $\Theta_j$
   - Compute the cost $C_\Theta$ (weighted consensus) of $\Theta_j$ using the MSAC cost function
   - if $C_\Theta > C^\bullet$ then $C^\bullet \leftarrow C_\Theta$, $\Theta^\bullet \leftarrow \Theta_j$
```

The algorithm 4 shows how WSAC works and figure 5 shows a graphical example. The first local search with point of reference $l = 1$ for the sliding window S_l is represented by figures 5(a), 5(b), 5(c) and 5(d). The first local search was not successful. The second local search, with $l = 4$, is represented by the figures 5(e), 5(f), 5(g) and 5(h). In Figure 5(h) we found a line supported by points 6, 7, 8 and 10. In the global search, points 12 and 13 are added to the set S_Θ, and the line is putted into the map. Finally the last local search with $l = 4$ and the remaining points will be unsuccessful (see Figure 5(i)).

4 Experimental Results

We perform two kind of tests: 1) using simulated data of a Laser Range Finder mounted on a mobile robot in a structured environment, and 2) using a real mobile robot equipped with a LMS209-S02 SICK Laser Measurement System (see Figure 6(a)).

4.1 Test Using Synthetic Data

The simulated environment, shown in figure 6(b), has 12 walls, each one labeled with a number. The aim of this test is to evaluate the robustness of the proposed

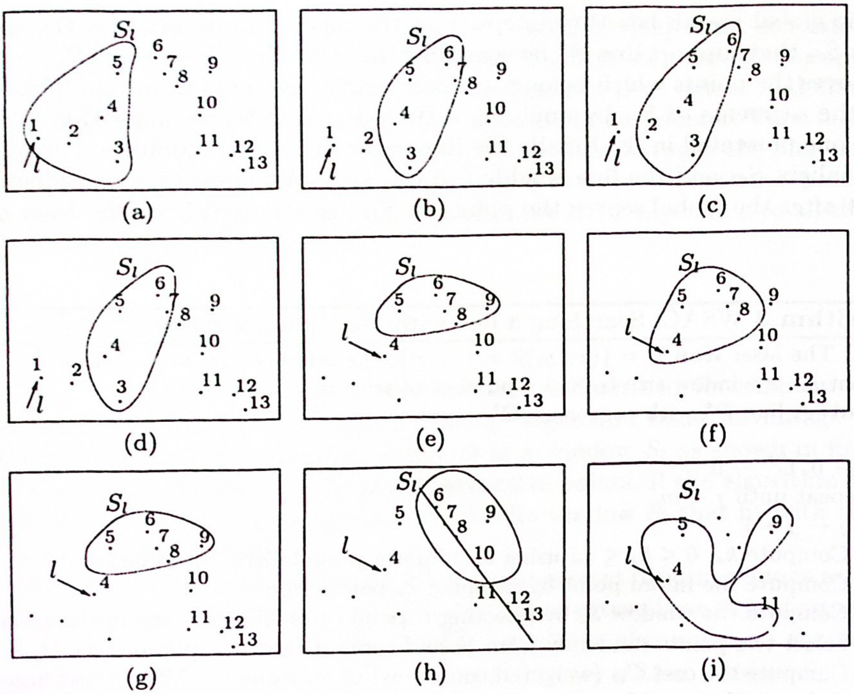
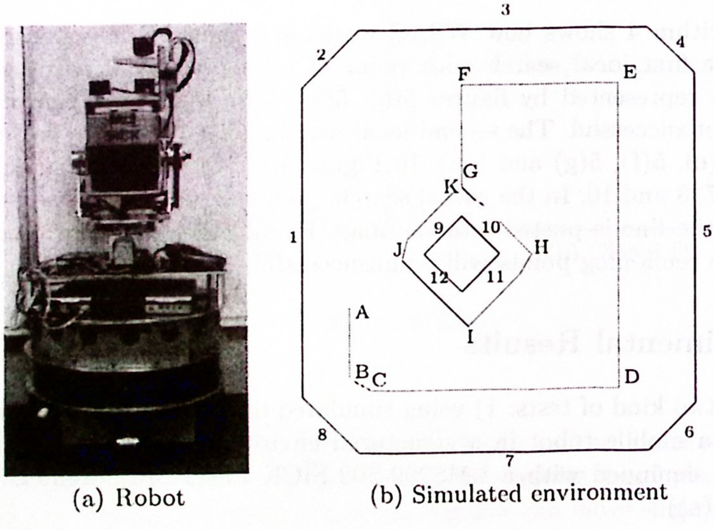


Fig. 5. An example using WSAC



(a) Robot (b) Simulated environment

Fig. 6. Our mobile robot and a simulated environment

Algorithm 4 WSAC

Input: The laser scan $\mathcal{P} = \{(x_i, y_i)|i = 1 \ldots n\}$
Output: A map of lines $\mathcal{M}$

```
1. $l \leftarrow 1$
2. while $l < n$ do
   (a) Perform the local search (algorithm 3) obtaining $\Theta$, $S_\Theta$ and $C_\Theta$
   (b) if $C_\Theta < C_{min}$ then
       - Merge into $S_\Theta$ the points of $\mathcal{P}$ correctly represented by $\Theta$
       - Extract from $S_\Theta$ points which belongs to small segments
       - if $S_\Theta$ have more than 1 element, recompute the line parameters from $S_\Theta$,
       remove the points in $S_\Theta$ from $\mathcal{P}$ and add $\Theta$ into $\mathcal{M}$
       - Relocate $l$ to the next point in $\mathcal{P}$
   (c) else
       - Increment $l$ with a constant value $l \leftarrow l + k$
```

method against the popular SMSM method. In this experiment the robot follows the path ABCDEFGHIJK doing a total of 1000 synthetic laser scans.

Each laser scan has 361 measurements covering 180° with a resolution of 0.5° and a maximum distance of $32m$. The simulator replaces 20% of the measurements with a spurious noise (simulated by adding an uniform random value between 0 and the maximum distance). The remaining measurements were only contaminated with a Gaussian random noise with $\sigma = 3cm$.

Table 1 shows the results for this experiment, where the best values are in boldface. As we can see, WSAC got better results: in the parameters of the line ($|\Delta r|$ and $|\Delta\alpha|$) and in a better association of points with lines (shown in the last column of the table).

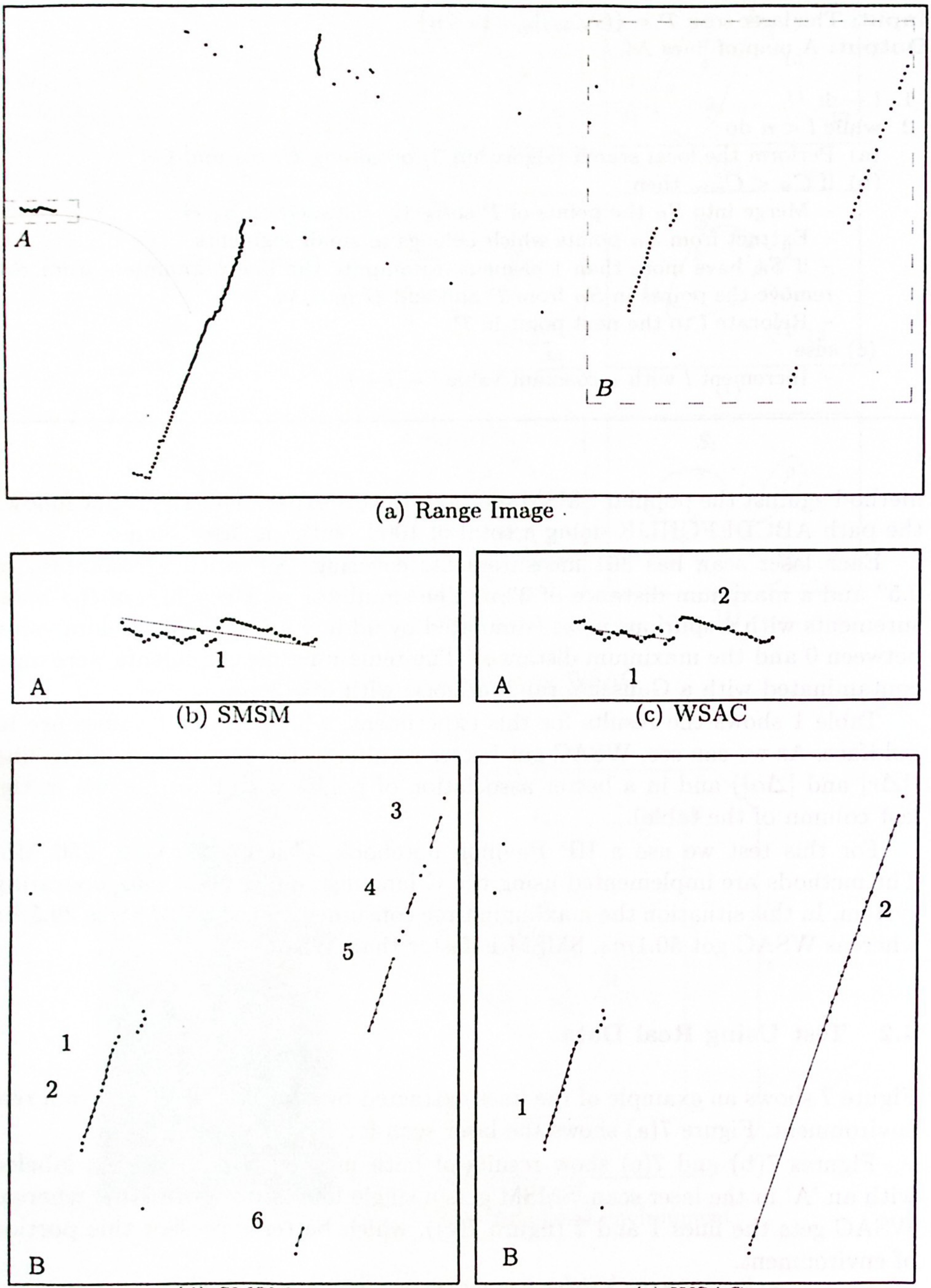
For this test we use a HP Pavilion notebook, Celeron 1.1 Ghz, 256 Mb. Thr methods are implemented using the C language under the Linux operating system. In this situation the maximum time consumed by the SMSM was $29.5ms$ whereas WSAC got $50.1ms$. SMSM is faster than WSAC.

4.2 Test Using Real Data

Figure 7 shows an example of the lines extracted by SMSM and WSAC in a real environment. Figure 7(a) shows the laser scan for this example.

Figures 7(b) and 7(c) show results of both methods over the area labeled with an 'A' in the laser scan. SMSM gets a single line (see Figure 7(b)) whereas WSAC gets the lines 1 and 2 (figure 7(c)), which better represent this portion of environment.

Figures 7(d) and 7(e) show the lines obtained by both methods over the area labeled with an 'B' in the laser scan. In this case, SMSM reported more segment lines than WSAC. However lines 1 and 2 in Figure 7(d) correspond to the single line 1 in the Figure 7(e). Similarly, lines 3, 4, 5 and 6 in Figure 7(d) correspond



(a) Range Image

(b) SMSM

(c) WSAC

(d) SMSM

(e) WSAC

Fig. 7. Results using the real mobile robot

Table 1. Results obtained in the simulated environment

	qty		$\lvert\Delta\alpha\rvert$		$\sigma_{\Delta\alpha}$		$\lvert\Delta r\rvert$		$\sigma_{\Delta r}$		$\overline{n_{\text{points}}}$	
line	SMSM	WSAC	SMSM	WSAC	SMSM	WSAC	SMSM	WSAC	SMSM	WSAC	SMSM	WSAC
1	**629**	609	0.017	**0.111**	0.027	**0.017**	25.433	**17.125**	39.96	**25.457**	35.11	**62.59**
2	**454**	422	0.030	**0.023**	0.028	**0.020**	52.682	**42.947**	50.090	**45.658**	12.18	**20.05**
3	**632**	626	0.011	**0.011**	0.020	**0.017**	18.386	**16.916**	31.302	**29.361**	40.70	**69.88**
4	**468**	393	0.036	**0.029**	0.029	**0.026**	57.801	**46.617**	56.035	**46.758**	11.53	**21.59**
5	**692**	689	0.011	**0.009**	0.022	**0.014**	15.204	**13.405**	24.717	**22.942**	45.33	**76.20**
6	**392**	322	0.034	**0.029**	0.029	**0.027**	51.614	**42.923**	56.139	**49.473**	12.18	**22.02**
7	**566**	560	0.014	**0.010**	0.023	**0.016**	25.733	**16.336**	42.320	**27.521**	39.85	**70.57**
8	**312**	266	0.352	**0.026**	0.031	**0.026**	43.312	**32.764**	47.049	**40.455**	15.19	**31.43**
9	**237**	192	0.035	**0.022**	0.035	**0.025**	56.155	**36.307**	60.181	**43.705**	13.79	**31.32**
10	**314**	282	0.034	**0.026**	0.031	**0.027**	49.564	**40.639**	50.465	**50.192**	14.21	**26.95**
11	**241**	216	0.036	**0.026**	0.032	**0.026**	47.576	**40.583**	52.376	**56.977**	14.44	**29.70**
12	**155**	131	0.043	**0.027**	0.037	**0.027**	42.248	**23.958**	47.760	**29.220**	17.83	**39.51**

to the single line 2 in Figure 7(e). Also in this case, WSAC got better results than SMSM.

5 Conclusions

We propose a robust method to find multiple lines in a laser scan, avoiding problems due to outliers and merging local and global strategies. We use a M-Estimator within a RANSAC method to find a line from a short sequence of points of the laser scan (the local strategy), then the line is evaluated and refined using the whole set of points (the global strategy), discarding those point belonging to very small segments.

This method is fast and it is able to find better results than the SMSM method, a very well known method.

In the future we plan to formulate a global cost function to evaluate how good is the set of lines found. With this cost function, inserting or deleting lines will depend on how they affect the global cost.

References

1. Geovany Araujo Borges and Marie-José Aldon. Line extraction in 2d range images for mobile robotics. *J. Intell. Robotics Syst.*, 40(3):267–297, 2004.
2. Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* Wiley-Interscience, 1976.
3. Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
4. David A. Forsyth and Jean Ponce. *Computer Vision: A Modern Approach.* Prentice Hall, 2003.

5. B. Giesler, R. Graf, R. Dillmann, and C.F.R. Weiman. Fast mapping using the Log-Hough transformation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1998.
6. F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel. *Robust Statistics: The Approach Based on Influence Functions.* John Wiley and Sons, New York, N.Y, 1986.
7. P.V.C. Hough. Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*, 1959.
8. Y. Liu, R. Emery, O. Chakrabartl, W. Burgard, and S. Thrun. Using EM to learn 3D models with mobile robots. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2001.
9. P. Meer, C. V. Stewart, and D. E. Tyler. Introduction, robust computer vision: An interdisciplinary challenge. *Computer Vision and Image Understanding*, 2000.
10. I. Miller. *John E. Freund's mathematical statistics.* Prentice Hall, 6 edition, 1999.
11. Viet Nguyen, Agostino Martinelli, Nicola Tomatis, and Roland Siegwart. A comparison of line extraction algorithms using 2d. laser rangefinder for indoor mobile robotics. In *International Conference on Intelligent Robots and Systems (IROS 2005)*, 2005.
12. P J Rousseeuw and A M Leroy. *Robust regression & outlier detection.* John Wiley & Sons, Inc., New York, NY, USA, 1987.
13. D. Sack and W Burgard. A comparison of methods for line extraction from range data. In *Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicle (IAV)*, 2003.
14. Stephen M. Stigler. *The History of Statistics: The Measurement of Uncertainty Before 1900.* Harvard University Press, Cambridge, Massachusetts, 1986.
15. P H S Torr and A Zisserman. Robust computation and parameterization of multiple view relations. In *ICCV6*, pages 727 – 732, 1998.
16. Carl F.R. Weiman. Application of log-hough transform to lidar navigation, phase i sbir final report. Technical report, HelpMate Robotics Inc., 1994.
17. Victor J. Yohai. Regresión robusta. CEMA Working Papers 9, Universidad del CEMA, December 1979. available at http://ideas.repec.org/p/cem/doctra/9.html.
18. Xu Zezhong, Liu Jilin, and Xiang Zhiyu. Map building and localization using 2d range scanner. In *Proceedings 2003 IEEE International Symposium on Computational Intelligence in Mobile Robotics and Automation*, pages 848–853, 2003.