Global Localization of Mobile Robots for Indoor Environments Using Natural Landmarks

Sergio F. Hernandez-Alamilla
Intelligent Systems Laboratory
ITESM Campus Cuernavaca
Cuernavaca, Morelos, Mexico
sergio.alamilla@gmail.com

Eduardo F. Morales
Department of Computer Science
INAOEP
Tonantzintla, Puebla, Mexico
emorales@inaoep.mx

***Abstract*— This paper introduces a global localization system based on natural landmarks for indoor mobile robots. The proposed approach is based on recognition of natural landmarks from laser scanner data. A previously built grid-based map is pre-processed off-line to obtain a model of landmarks and their attributes for each cell. The robot's position and orientation are calculated by finding correspondence between the identified landmarks from robot's current position and the landmarks associated to the model. This proposed approach called GL2 follows a two stage process. Initially a fast initial filter based on the number and type of landmarks is used to substantially reduce the search space. The second stage uses a modified discrete relaxation algorithm to perform a more detailed analysis and find the robot's location and orientation. It is shown the robustness of the algorithm in complex and real office like environments, even in the presence of previously unknown obstacles.**

***Keywords*—global localization, natural landmarks, grid maps**

I. Introduction

The ability for mobile robots to locate themselves in their environment is not only a fundamental problem in robotics but also a pre-requisite to many tasks such as navigation. There are two types of localization problems: local localization and global localization. Local localization techniques aim to compensate for odometric errors during navigation and require information about the initial position of the robot. The vast majority of existing localization algorithms address this problem (see e.g., [1] for a review). A more difficult task is the global localization problem where the aim is to locate the robot's position without prior knowledge of its current location. Even more difficult is the kidnapped robot problem [4], in which a robot is moved to another place without being told. These problems are particularly hard in dynamic environments where new obstacles can corrupt the robot's sensor measurements [5]. This work is focused on global localization using natural landmarks. The general idea consists of pre-processing a previously known grid-based map to obtain a model of visible landmarks and their attributes for each cell. The global localization problem is then reduced into finding correspondence between the landmarks associated to a cell and the landmarks observed by the robot from its current position. The proposed approach called GL2, follows a two stage process. During the first stage a fast initial filter is used to eliminate a large set of possible candidates. This filter is based on the number and type of landmarks observed by the robot from its position. The second stage consists on a more detailed comparison to eliminate most of the remaining candidates. This detailed correspondence is determinated using a modified discrete relaxation algorithm based on the spatial distribution of landmarks. It is shown the robustness of GL2 in complex and real office like environments, even in the presence of new obstacles. This paper is organized as follows. Section 2 describes some previous related work. Section 3 describes the specific kind of natural landmarks used in this work and how to identify them. Section 4 describes how a grid-based map is pre-processed to obtain a model based on landmarks. In Section 5, the global localization algorithm proposed in this paper is introduced. Section 6 describes some experiments and results with the proposed method and Section 7 gives conclusions and future research directions.

II. Related Work

Many authors have used information on cells trying to solve the global localization problem. Due to the difficulty to compare the observations with all the cells, in [9] the authors suggested to use a sampling mechanism guided by a similarity function. They found, however, some problems in defining a monotonically increasing similarity function and an adequate sampling mechanism. Several recent methods have been proposed in the literature for the global localization problem. The most promising ones share the same mathematical basis and use Markov localization algorithms with filter conditions (e.g., [10]), multi-hypotheses Kalman filters which represents beliefs using mixtures of Gaussians (e.g., [12]) or Monte Carlo localization algorithms (e.g., [11]). The former two assume Gaussian noise and require parametric models, while the later can represent multi-modal probability distributions, however, it is difficult to estimate the number of particles to use. All of these methods are probabilistic and require the robot to move around while the probabilities converge towards one localized peak. In this paper, a simple, yet robust and effective method is proposed that can deal with real office like environments. GL2 can effectively deal with noise and new obstacles not included in the original map. In the following section a natural landmark

1–4244–0025–2/06/$20.00 © 2006 IEEE

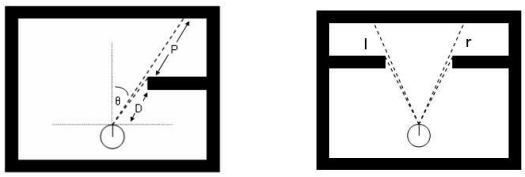
extraction algorithm is presented.

III. Natural Landmarks

Different natural landmarks have been proposed in the literature to find the robot's position. In [6], walls are used as natural landmarks, obtained through the Hough transform. In [6] a detector of concave corners is used, which are extracted from the data of a laser sensor. Similarly, in [7], discontinuities are used to solve the problem of local localization. In this case we use a laser sensor: the Sick-LMS200 which obtains the distance to the closest obstacle over a plane parallel to the floor. Precision is of ±5mm in distance measured every 1, 0.5 or 0.25 degrees. In this work we use discontinuities, corners and walls as landmarks to solve the global localization problem. Combining these three types of landmarks gives the robot the capacity to identify enough landmarks from any position in a wide variety of environments. Another advantage of the approach is that landmarks can be easily extracted from laser scanner data with high accuracy. Natural landmarks are identified in three steps: the first step identifies discontinuities and based on them the whole laser range is divided into segments. The second step identifies corners from the segments by dividing them in sub-segments. The third step takes the remaining segments and performs a fast local Hough transform to calculate the parameters associated with possible walls. By identifying landmarks from the simplest to the more complex the identification process is simplified. Each landmark (discontinuity, corner or wall) is associated with a set of distinctive attributes. This process is described as follows.

A. Discontinuities

A discontinuity is defined as an abrupt variation in the measured distance of two consecutive readings of the laser, as shown in Figure 1.



(a) Attributes of a discontinuity

(b) Type of discontinuity (attribute T).

Fig. 1. An example of a discontinuity

As shown in Figure 1(a) a discontinuity can be associated with different attributes: the *distance* (D) between robot and discontinuity, the *depth* (P) or difference between two contiguous readings that caused the discontinuity, the *angle* (θ) of the discontinuity with respect to the front of the robot, and the *type* (T) that determines if a discontinuity is left *(l)* or right *(r)* oriented according to the location of the non visible region (see Figure 1(b)). Once a set of N_d discontinuities has been identified the whole laser range is divided into $N_d + 1$ segments. This corresponds with the first level of the segment's tree in Figure 2.

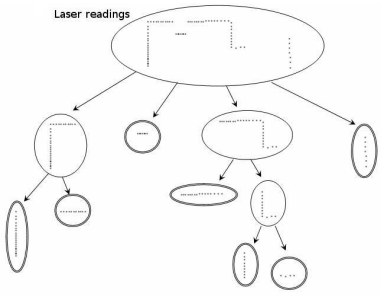


Fig. 2. Tree of segments after identify discontinuities and corners.

B. Corners

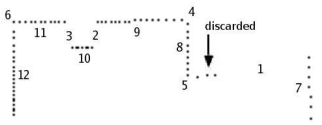
Each segment is considered in order to try to identify corners. The farthest point in the segment from an imaginary line passing by the initial and final point of the segment is obtained. If the distance between this point and the imaginary line is greater than a threshold μ_e then it is considered as a corner. The segment is divided using the corner and the same process is performed on each of the sub-segments. As shown in Figure 2, at the end of the process, segment division can be seen as a tree of segments. Leafs of the tree are possible walls (see Figure 2).

C. Walls

Each leaf of the tree is taken and a fast local Hough transform if performed in order to calculate the parameters of the corresponding lines. Walls are lines in laser readings. This local Hough transform is fast because there is only one possible line in each segment. This avoids using clustering techniques which are necessary to find local maximums when it is possible to find more than one line. A natural landmark, m_i, can be represented by a tuple of four attributes: (D, θ, A, T). D and θ are the distance from the landmark to the robot and the orientation of the landmark relative to the robot respectively. T is the type of the landmark; *l* for left discontinuity, *r* for right discontinuity, *e* for corner and *p* for walls. A is a distinctive attribute and its value depends on the type of the landmark; for discontinuities A is depth (*P* in Fig 1), for corners A is concave (*cc*) or convex (*cx*) and for walls A is the length of the wall. Figure 3 shows an example of natural landmarks identified by GL2. In the following section, the pre-processing of a grid-based map in order to obtain a model of the environment based on landmarks is presented.

IV. Pre-processing of a grid-based map

The global localization algorithm presented here is part of a complete robotic system. There are other modules like planning, navigation and map building. The map building module [15] exports a grid-based map so in this work we assume that a grid-based map of the environment has been previously built by the robot. Each cell in the map is associated with an occupancy probability. The pre-processing stage only



(a) Laser readings.

No	D	θ	A	T	Type of landmark
1	2.18	-73.0	1.41	l	left discontinuity
2	1.10	-6.0	0.40	l	left discontinuity
3	1.11	10.0	0.41	r	right discontinuity
4	2.17	-47.0	cc	e	corner
5	1.71	-68.0	cx	e	corner
6	1.84	36.0	cc	e	corner
7	3.46	-89.50	1.06	p	wall
8	1.59	-89.75	0.85	p	wall
9	1.49	0.0	1.36	p	wall
10	1.10	0.0	0.39	p	wall
11	1.50	0.0	0.80	p	wall
12	1.10	89.75	1.43	p	wall

(b) Set of identified natural landmarks.

Fig. 3. Identification of natural landmarks

needs to know if a cell is occupied or not which can be determined with a threshold value. The objective of the pre-processing stage is to associate a set of landmarks and their attributes to each cell. We use ray tracing to simulate the laser readings and then extract the landmarks from the simulated readings. A range of $360°$ is considered for each cell. At the end of this process each cell c_i of the map is associated with a set C_i which contains all the landmarks "visible" from that cell. The attribute θ is relative to the coordinate system of the map; this is important because after the position of the robot has been determined, the attribute θ will be used to calculate the orientation of the robot relative to the map.

As expected, the size of the cells have a direct impact on memory and computational requirements of the system and on the magnitude of the error in the position and orientation of the robot. Larger cells produce larger localization errors but smaller memory requirements, while smaller cells produce more accurate position estimates but require more computational power.

V. Global localization method

The objective of the localization method is to determine the position and orientation of the robot in a map. The first step consists of obtaining the landmarks that the robot observes at $360°$. In our experiments we used a Nomad Scout II mobile robot equipped with a laser range finder Sick LMS200 considering an angular resolution of $0.5°$. Since this sensor obtains the distance and angle of the nearest obstacles within a range of $180°$, the landmarks are obtained in two steps. First the landmarks set C_1 is extracted from the current position of the robot, the robot is rotated $180°$ and another set C_2 of landmarks is extracted. Obviously it is not easy to exactly rotate $180°$ degrees. An algorithm of pose tracking which is also part of the whole robotic system is used to turn precisely [16]. The final set of observed landmarks C_o is the

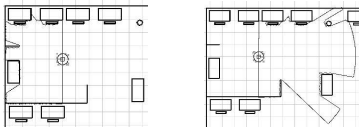


Fig. 4. The complete $360°$ landmark set is extracted in two steps due to the limited visibility range of the sensor we have (Sick LMS200).

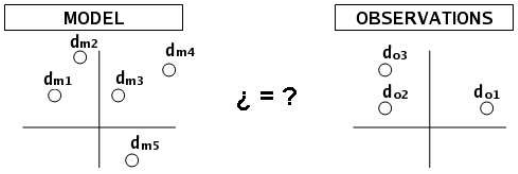


Fig. 5. The problem of finding the correspondence between the model and observations of the robot.

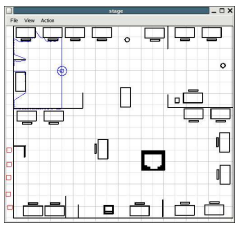
union of C_1 and C_2 as shown in Figure 4. From here on $C_o = (d_{o1}, d_{o2}, ..., d_{ok})$ will refer to the set of k landmarks observed by the robot and $C_m = (d_{m1}, d_{m2}, ..., d_{mj})$ will refer to the set of j landmarks associated with a cell of the model.

The information stored in each cell by the pre-processing stage is used to calculate the robot's position. As illustrated in Figure 5, this process consists of matching observed landmarks (C_o) with landmarks in the stored model (C_m). In many cases there is no exact match due to noise in the sensors measurements or changes in the environment. In real office like environments it is common for people to move objects, such as chairs or tables, from their original position. In Section V-B a discrete relaxation algorithm is presented to deal with this matching problem.

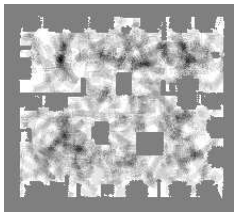
Searching through all the cells associated with the map and matching their landmarks with the landmarks observed by the robot would require considerable computation power. To simplify this process, GL2 follows a two stage process. The first stage gets rid a large number of candidate positions with a filter based on the number and type of landmarks. The second stage performs a more detailed analysis on the remaining cells, based on spatial distribution of the landmarks. These two stages are described in the following sections.

A. Initial Filter

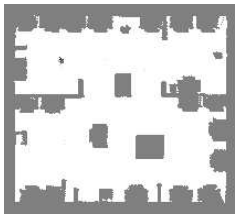
The initial filter counts the number of landmarks obtained from the laser readings that match the distance, orientation, distinctive attribute and type of the previously stored landmarks associated to each cell. A tolerance of 10 centimeters for the distance and depth of each discontinuity is employed for the search process. This last consideration is not due to the imprecision of the sensor but to the map. The number of landmarks with similar attributes to the observed ones is divided among M, with M the total of landmarks associated to each cell. Figure 6(b) shows the result of applying this initial filter to the map given the landmarks observed by the robot; the darker regions represents the higher percentage values and consequently the most probable cells for the robot's location.



(a) Simulated environment.



(b) After the first filter.



(c) After discrete relaxation.

Fig. 6. Results from the first filter applied to a simulated environment.

B. Algorithm of discrete relaxation

From the initial filter only those cells that have half or more of the larger percentage are considered in the second phase. In this paper, a modified discrete relaxation algorithm is used to determine the similarity of each cell with the observations of the robot.

Consider the matching problem shown in Figure 5. By itself the discrete relaxation algorithm can be very inefficient if all the possible assignments are considered. However, using the information of attributes associated to the obtained landmarks in the pre-processing stage, only a small subset of possible matches, usually 2 or more, are considered, which considerably reduces the complexity. An example of this is shown in Table I(a): the initial possible matchings are initialized with 0 while the rest are not considered. This happens because in practice it is very difficult to find two landmarks with the same values in their attributes. To obtain more accurate matches than those obtained during the first stage, we also consider the Euclidean distance and the angular difference between landmarks in this stage.

In Table II the modified discrete relaxation algorithm is presented. At the end of the discrete relaxation algorithm, the maximum value of each line in matrix A has the number of matches of the i-th observed landmark. In Table I(b) the maximum of each row gives the matchings of landmarks between set C_o and C_m. Obviously the maximum of each row can be at most $i-1$ for i observations. In Table II $R(P_i, P_j)$ denotes a relation between two landmarks, this relation is the Euclidean distance and the angular difference, the $\approx$ operator is used because distance and angle measure can not be strictly equal due to sensor and map imprecision, consequently a threshold is used. A description of the basic discrete relaxation algorithm can be found in [8].

TABLE I

EXAMPLE OF DISCRETE RELAXATION ALGORITHM.

	d_{m1}	d_{m2}	d_{m3}	d_{m4}	d_{m5}
d_{o1}	-	-	-	0	0
d_{o2}	0	0	0	-	-
d_{o3}	-	0	0	-	-

(a) Initial possible matchings.

	d_{m1}	d_{m2}	d_{m3}	d_{m4}	d_{m5}
d_{o1}	-	-	-	0	2
d_{o2}	2	0	1	-	-
d_{o3}	-	2	1	-	-

(b) After discrete relaxation.

TABLE II

THE DISCRETE RELAXATION ALGORITHM.

Let $P_i, i = 1 \ldots N$ be the observed landmarks
Let $S(P_i), i = 1 \ldots M$ be the set of initially compatible labels
Let $A[N][M]$ be a matrix of counters
procedure *Discrete_Relaxation*(P,S) {
repeat {
 for each($P_i, S(P_i)$) do {
  for each landmark Lk in $S(P_i)$
   for each relation $R(P_i, P_j)$ in the observations
    if $\exists Lm \in S(P_j)$ in the model with $R(Lk, Lm) \approx R(P_i, P_j)$
    then $A[Pi][S(Pi)] = A[Pi][S(Pi)] + 1$
  }
 } until $i = N$;
}

Figure 6(c) shows the result after applying the discrete relaxation algorithm to the map shown in Figure 6(b). The algorithm of discrete relaxation does not need to find an exact correspondence with each observation, and the geometric restrictions among landmarks play a relevant role in the matching process. At the end of this stage only a few cells remain, all of them together. It is clear that these cells have the same number of landmarks matching with observations. To choose only one of them a least-square criterion is used.

C. Least-square criterion

At this point only a small number of cells remain, all of them have the same number of landmarks and the same matchings. To decide which of them corresponds with the position of the robot a least-square criterion is used. For this purpose we take advantage of the attribute D of landmarks (distance from the robot to the landmark). The best adjustment of observations C_o with the model is obtained by calculating the square of the difference between the value of attribute D on each corresponding match. In Equation 1, N is the number of matchings, $E(c_j)$ is the adjustment error of the j-th cell c_j; D_{oi} and D_{mi} are the attribute D of landmarks d_{oi} and d_{mi} respectively.

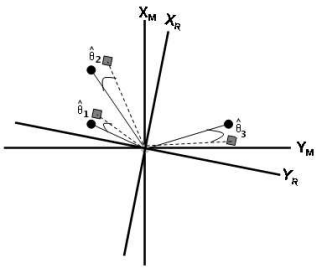


Fig. 7. Orientation of the robot relative to the map

$$E(c_j) = \sum_{i=1}^{N} (D_{oi} - D_{mi})^2 \quad (1)$$

In this way the cell c_k that corresponds with the position of the robot is:

$$c_k = argmin(E(c_j)) \quad (2)$$

D. Determining orientation of the robot

To calculate the robot orientation given the robot position is quite simple. Let d_{oi} be a landmark in the observations and d_{mi} a landmark in the model of the final position cell. For each landmark match (d_{mi}, d_{oi}), the arithmetic difference is calculated between the attribute θ_{oi} of d_{oi} and the attribute θ_{mi} of d_{mi}. Remember that θ_{mi} is the orientation of a landmark relative to the map's coordinate system and θ_{oi} is the orientation of a landmark relative to the robot's coordinate system. The final orientation is the average of all calculated differences. In Figure 7 $\hat{\theta}_i$ is the i-th difference between orientation in observations and orientation in the model, X_M, Y_M is the Cartesian coordinates system of the model and X_R, Y_R is the Cartesian coordinates system of the robot, small circles and small squares represents d_{mi} and d_{oi} landmarks respectively. In Equation 3, N is the number of matchings, θ_{oi} and θ_{mi} are the attribute θ of d_{oi} and d_{mi} respectively.

$$\theta_R = \frac{\sum_{i=1}^{N} \hat{\theta}_i}{N} \quad where \quad \hat{\theta}_i = \theta_{oi} - \theta_{mi} \quad (3)$$

VI. Experiments and Tests

We performed several experiments with GL2 to different office-type environments and placing the robot in different positions. We emphasized in our tests different places which could have similar characteristics in terms of landmarks with other positions in the maps, with very encouraging results. The maps used in the tests have some imprecision as they were build with a laser sensor by the robot. The size of the cells is 0.05 m. To consider more realistic conditions, we also added new objects to the environment at different places after the construction and pre-processing of the map.

Each new obstacle can represent a person or additional furniture obstructing some of the original landmarks and creating new ones. Figure 8 shows the performance on GL2 with 5 new obstacles. As can be seen from the figure, the initial filter finds several probable locations for the robot, however the discrete relaxation algorithm continues to successfully locate the position of the robot.

TABLE III

Average performance of GL2 for 30 random positions on 3 different simulated environments

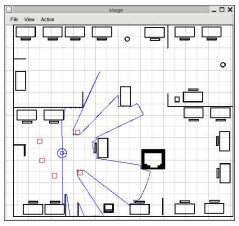
Map	No. cells	Time stage 1 (secs.)	No. cells stage 2	Time stage 2 (secs.)	Memory used (Mb)	Loc. success (%)
Map1	480x580	1.7	2,246.5	0.3	18.66	100
Map2	340x400	0.509	790.8	0.164	3.93	100
Map3	280x280	0.176	297.1	0.130	0.8	100

TABLE IV

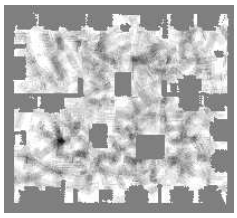
Average performance of GL2 for 30 random positions on 2 different real environments

Map	No. cells	Time stage 1 (seg.)	No cells stage 2	Time stage 2 (seg.)	Mem. used (Mb)	Loc. success. (%)
Offices	250x400	0.645	567.3	0.279	14.93	96.6
Laboratory	190x170	0.398	1120.2	0.180	7.3	100

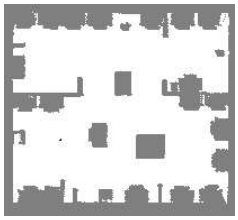
Table III has the performance times of GL2 with 3 different maps on simulation. Map2 is the one shown in Figure 8(a) of approximately 10×10 meters. Map1 is twice as big as Map2 (20×20 meters), while Map3 is half of its size (5×5 meters). Table III includes the number of cells, the average time used in the first filter process for 30 random positions, the average number of cells used in the discrete relaxation algorithm, the average processing time for the discrete relaxation algorithm and the memory space required to save the model. Table IV has the performance times of GL2 with 2 different real maps. Laboratory map is the one shown in Figure 9 of approximately 8×8 meters. The map was built with the map building module in the Intelligent Systems Laboratory of the Campus. Offices map is the one shown in Figure 10 of approximately 15×8 meters. It corresponds to offices of staff members of the Campus. It is a challenging environment because of the open/close doors and mobile furniture. Table IV includes the number of cells, the average time used in the first filter process for 30 random positions, the average number of cells used in the discrete relaxation algorithm, the average processing time for the discrete relaxation algorithm and the memory space required to save the model. The 96.6% in localization success in Table IV is because one of the experiments was not successful. The problem is associated with similar regions of the map, it happens because sometimes there are more than one region in the map that matches with the landmarks observed by the robot. This situation is difficult to find in real environments due to the variety of landmarks used in this work. Note that even in Map1 the memory space required is small in comparison with the capacity of actual computers. All tests were run with an AMD Athlon laptop at 1.53 MHz, 192Mb of memory. For simulation we use the software of the Player/Stage project which is focused on robot and sensor applications. Further information about player/stage can be found in [14].



(a) Additional objects added



(b) Most probable cells after first filter



(c) Final position found

Fig. 8. Experiment with new objects added to the environment

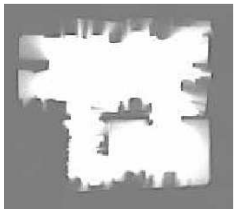


Fig. 9. Map of the Intelligent Systems Laboratory of approximately 8×8 meters.

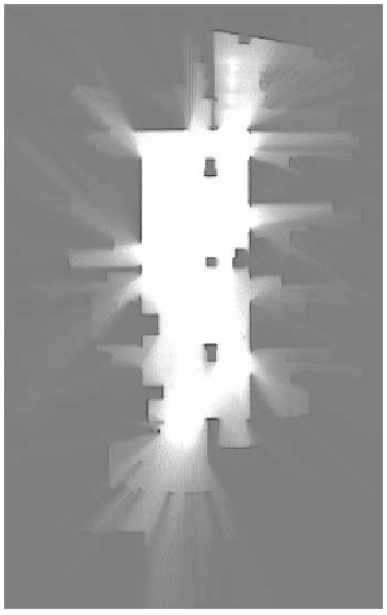


Fig. 10. Map of the offices of aproximately 15×8 meters.

VII. Conclusions and Future Work

Mobile robot localization has been estimated as the most fundamental problem to provide autonomous capabilities to a mobile robot [13]. In this paper, we have described a global localization algorithm which shows good performance in complex office environments in the presence of noise and new obstacles. The algorithm is based on detecting natural landmarks and obtaining their attributes, and follows a two stage process. The first stage eliminates a large number of possible candidates, while the second stage performs a more detailed refinement process. There are several research directions that are worth exploring. In particular, we would like to improve the robustness of the system in cases where there are few landmarks by incorporating another type of natural landmarks. The precision of the localization process is proportional to the size of the cells in the grid map, while the processing time and memory requirements are inversely proportional to the size of the cell.

References

[1] J. Borenstein, B. Everett, and L. Feng. Navigating Mobile Robots: Systems and Techniques. A. K. Peters, Ltd., Wellesley, MA, 1996.

[2] J.S. Gutmann and C. Schlegel. Amos: Comparison of scan matching approaches for self-localization in indoor environments. In *Proceedings of the 1st Euromicro Workshop on Advanced Mobile Robots*. IEEE Computer Society Press, 1996.

[3] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In I.J. Cox and G.T. Wilfong, editors, *Autonomous Robot Vehnicles*, pg. 167 193. Springer-Verlag, 1990.

[4] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11:391 427, 1999.

[5] W. Burgard, A.B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2):3 55, 1999.

[6] N. Tomatis. Hybrid, Metric - Topological Robot Navigation. Politecnico de Lausanne, Lausanne EPFL, 2001.

[7] Z. Xiaowei, H.Y. Khing, C.C. Sen, and Z. Yi. The Localization of Movil Robot Based on Laser Scanner. School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 2000.

[8] G. Stockman and A Goshtasby, 2-D and 3-D Image Registration: A Tutorial, *Computer Vision and Pattern Recognition 2004*, http://www.cse.msu.edu/˜stockman/RegTutorial/, Washington DC.

[9] P. MacKenzie and G. Dudek. Precise positioning using model-based maps. In *Proc. Int. Conf. Robotics and Automation*, pp. 1615-1651. San Diego, CA, 1994.

[10] S. Thrun, M. Bennewitz, W. Burgard, A.B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz. Minerva: A second generation museum tour-guide robot. In *Proceedings of IEEE International Conference on Robotics and Automation* (ICRA'99), Detroit, Michigan, May 1999.

[11] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (CVPR'99), Fort Collins, CO, June 1999.

[12] S.I. Roumeliotis and G.A. Bekey. Bayesian estimation and kalman filtering: A unified framework for mobile robot localization. In *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA), pages 2985 2992, San Francisco, CA, 2000. IEEE.

[13] I.J. Cox., Blanche an experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Transactions on Robotics and Automation*, 7(2):193 204, 1991.

[14] B. P. Gerkey, R. T. Vaughan and A. Howard, The Player/Stage project, 2003, *http://playerstage.sourceforge.net* .

[15] V. Jaquez, Simultaneous localization and map building using particle filter. *Masters thesis*, ITESM Cuernavaca Mexico, 2005.

[16] S. Hernandez, Mobile robot navigation based on natural landmarks. *Masters thesis*, ITESM Cuernavaca Mexico, 2005.