

Robot Motion Planning for Map Building

Benjamín Tovar, Rafael Murrieta-Cid* and Claudia Esteves

ITESM Campus Ciudad de México, Calle del puente 222, Tlalpan, México D.F.
{betovar,rmurriet,cesteves}@campus.ccm.itesm.mx

Abstract

The goal of this work is to develop techniques that allow one or more robotic observers to operate with full autonomy while accomplishing the task of model building. The planning algorithm operates using certain simple but flexible models of the observer sensor and actuator abilities. We provide techniques that allow us to implement these sensor models on top of the capabilities of the actual (and off-the-shelf) sensors we have. It is worth keeping the following points in mind regarding our goals: 1) even with completely idealized sensing and mobility capabilities, the algorithmic task of model building is quite challenging. 2) computational techniques can be used to approximate and implement these idealized sensors on top of actual sensors. 3) the quality and success of the generated plans depend significantly on the observer capabilities; study of this dependency terms of high-level parameters describing the sensors (e.g., max. distance sensed, viewing frustum) is part of this work.

1 Introduction

In this section we analyze the generation of motion strategies for building a map of an indoor environment using a mobile robot with range and video sensors. The main problem to solve is to choose where the robot should move to get the next sensor reading. One concern of this study is the need to satisfy perception constraints while planning motions. We focus on the fundamental motion planning problem considering information provided by sensors. Some of the questions that this work tries to answer are: Which locations must be visited by a robot to efficiently map a building?, how must a robot move to explore an environment? To answer these questions we propose randomized motion planning techniques which take into account both geometric and image analysis computation. We describe a *planner* that selects the next position from a set generated with uniform probability. The selection of the next position is based on the maximization of a *utility function*.

*This research was funded by CONACyT, México

The evaluation of this function uses the concept of *robot information space*. Analyzing this information space, the utility function selects the next position using the following criteria: 1) Trajectories where the robot may identify objects (landmarks) that can be used to navigate are preferred; 2) The robots localization uncertainty should be minimized; 3) The number of sensing operations should be minimized; 4) Energy consumption should be minimized by exploring a minimum distance trajectory; 5) The sensor capabilities (maximum distance sensed, viewing frustum) should be taken into account to compute motion strategies; 6) Motions should be performed in such a way that the robot perceives non-explored regions.

The implemented planner combines geometric calculations with an intensive use of information obtained by perception algorithms, such as scene recognition. The final result of the exploration is a multi-representational map consisting of polygons and landmarks, and including a *road-map* constructed from the trajectory followed by the robot.

Motivation Representing and understanding the environment from sensor readings is a fundamental task for robot navigation and exploration of an unknown environment. The approach proposed here is useful for these tasks. Also, we claim that this environment representation can be a crucial step toward building robots that can automatically achieve visual tasks, such as finding and tracking a target. Unlike the simpler “Go from A to B” task, these tasks require reasoning about both motion and visibility obstructions. The representation proposed is a complete and reliable map for these tasks because it gives a way to easily compute visibility and robot localization, which are a basic input to compute, motion strategies based on sensor information such as:

- Finding an evasive target requires one or several robots to sweep the environment so that the targets does not eventually sneak into an area that has already been explored. The planner can use

the representation proposed here to compute a robot motion such that, for any point p along this path, the part of the environment that has already been explored before reaching p is fully separated from the unexplored one by the region from the point p [8].

- In the target tracking task, the robot must visually track a target that may try to escape its field of view, for instance, by hiding behind an obstacle. An online planner can use the representation proposed here to decide how the robot should move [12, 17]. At each step, it computes the visibility region of the robot at several sample locations picked in a neighborhood of its current location, identifies the one that is most likely to contain the target, and commands the robot to move there.

2 Previous Work

Automatic model building is an important problem in mobile robotics [3, 7, 20, 6]. Several types of models have been proposed, i.e., topological maps [4], occupancy grids [7] which uses a 2D array to represent the environment. In this model, each cell is valued as free space, occupied space or unknown space. Grid-based building algorithms prove to be very simple and quite useful methods for obstacle avoidance and planning purposes [7]. However, when the size of the environment is big, it becomes hard to handle this type of models. 3-D models [21] or feature-based maps [2] and polygonal representation [3] have also been proposed. Feature-based models are another way to represent the environment by using geometric primitives. The most popular geometric primitive is the segment, which can be extracted from ultrasonic data [5], laser range-finder data [9], or vision data [1, 15, 16].

Most of these researchs have focused on developing techniques to extract relevant information from raw data and to integrate the collected data into a single model; a robot motion strategy is, however, not developed. In this work, we deal mainly with this problem. In [11] a map building motion planning strategy is presented. This work has shown that it is possible to find a function that reflects intuitively how the robot must explore the space. In a simple scheme, the evaluation function must assign a greater value to the position that best fits the compromise between possible elimination of unexplored space and energy consumption. One way to measure the size of the unexplored space is to measure the size of the free edge near it. A free edge is defined as the border between regions of explored and unexplored space, and a full edge as the border between explored

areas and obstacles. Energy has been measured by the distance that the robot must travel.

This strategy is based on the computation of the next-best view and the use of randomized motion planning. The concept of the next-best view is, however, almost purely based on geometric information, such as the visibility region from a robot position [18]. Uncertainty in robot localization and scene understanding are not taken into account. Our work tries to fill these gaps.

3 Our approach

The problem to solve in robot motion planning for map building is to determine a *good* motion strategy that allows the exploration of the whole environment, and to represent this new knowledge in such a way that not only the actual robot may deal with, but also the other mobile robots can work.

The definition of a *good* motion strategy depends on the desirable characteristics, such as minimum time, energy, uncertainty, representation complexity, etc. How to fit these requirements depends on the *explorer robot*, like sensors type and range, mechanical restrictions, etc. Until now most of the model building planners are based on pure geometric calculations. These planners generate simple and manageable representations, but contain limited environmental information.

Our work extends previous approaches by adding perceptual characteristics and the robots position uncertainty. The proposed criterion prefers those positions where the robot may recognize a *landmark* in order to perform landmark based navigation [13]. A landmark is defined as a recognizable object in the space. A simple definition of *place* is created where the robots location uncertainty is minimized. The *place* corresponds to the influence area of a landmark (or set of landmarks), i.e., the area from which the robot may see these entities.

Regarding perceptual features, the evaluation function should prefer those positions from which a perfectly recognizable object is found [14]. In this approach, it is proposed that the robot has online access to an object database, and the identification is made by using some classification technique, for instance Bayes rule. If the probability of an object belonging to any class is low, then the object must be taken only as a visibility obstacle.

The results of perception algorithms are known until the robot is at the next position, given that perceptual information can be obtained only if the robot is able to perceive the scene. For this reason, a two-step evaluation function is proposed.

The first step will evaluate the possible size of unexplored space, energy needed, distance to the nearest free edge, distance to nearest known walls and objects. All these parameters are inferred from the current robot position (by computing the visibility region [18]), which means that the robot does not have to go to every new possible position. The length of the free edges is used to estimate the size of the unexplored space. A very simple model of the uncertainty is used, where it grows proportional to the square root of the distance to travel. In this uncertainty model we also include a term where the rotation cost is calculated. It is preferred that the robot travels straight than rotating.

In the second step, the robot *actually* moves to the m best evaluated positions and selects the best one taking into account perceptual information. In a very simple way, it is as if the robot would take a glimpse from the threshold of several doors, and explore the room that it likes the best. The robot will stop exploring when there are no free edges to be visited left.

The perceptual information is composed by the identification probability of the objects perceived by the robot and some particular features such as corners, that allows the matching between the areas already explored. The utility function is constructed in such a way that it integrates all these features. The next possible position that maximizes the utility function is kept. The function will prefer positions combining proximity to the robot, proximity to a free edge, small uncertainty value of robot position, high object identification probability and ability to see features like corners. Positions near walls and objects will be discarded because many sensors become blind when the objects are very near.

The function is composed by terms representing each one of these measures mentioned above. It is proposed that the utility function has a multiplicative form. A position with a very low value on at least one of these measures will be discarded even though it may have a very good value on the others. For instance, a position very close to a free edge (with great chance of discovering new space) must be discarded if the robot has no information to integrate this new area to the explored space.

The utility function T is given by the following expression [19]:

$$T = (e^{(l_v - s - s_v)}) \left(\frac{e^{-|\theta|}}{\Gamma + \sqrt{s} + 1} \right) \left(\frac{1}{n} \sum_{j=1}^n p_j + e^{N_e} \right) f_{min}(d_l) \quad (1)$$

Where:

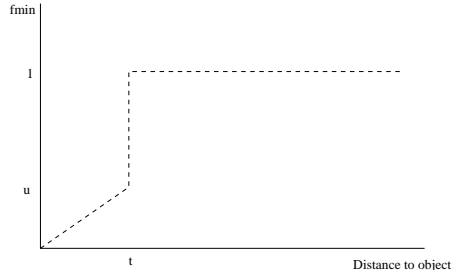


Figure 1: f_{min} function

| | |
|-----------|---|
| s | Distance from the robot to the next possible position |
| s_v | Distance from the next possible position to the closest free edge |
| l_v | Length of the closest free edge |
| θ | Needed orientation to reach the next robot's configuration |
| Γ | Cumulative uncertainty |
| p_j | Object identification probability |
| n | Number of landmarks inside a visibility region |
| N_e | Number of corners and end-points inside a visibility region |
| f_{min} | Function of the minimum distance from an object or full edge |
| d_l | Minimum distance from a full edge |

For f_{min} we choose a function like the shown in figure 1. Before a distance threshold t the function takes a low value, discriminating those positions near the objects. After the threshold the function takes the value of 1, to let the value of T reside on the other parameters.

The p_j is the identification probability for a landmark. A landmark is defined as a remarkable object. The landmark should have some properties that distinguish them from other objects, i.e.:

- **Discrimination.** A landmark should be easy to differentiate from other surrounding objects.
- **Accuracy.** A landmark must be useful to reduce the uncertainty of the robot position.

3.1 Landmark identification

Landmarks in indoor environments are often structures such as corners, doors, columns, posters, etc. To compute the distance between a landmark and the robot by using a single camera, we suppose that the landmark size is known. Our landmarks are posters, doors, and columns. To assume that for a given environment the size of these objects is known a priori is not unrealistic. An object is labeled as a landmark if and only if its object identification probability is

greater than a given threshold and has a remarkable shape (different from surrounding objects). The object shape can be obtained by using a segmentation algorithm as the one presented in [15]. Object identification is obtained by comparing vector features with a database composed of different classes, issued from a learning process. The database is a function of the type of environment. The object identification probability is calculated using Bayes rule.

3.2 Line-fitting

Since we have a laser range finder as our sensor, it is necessary to recover the lines that forms the actual visibility region from the points that the laser gives.

We generate polylines with the laser data obtained as an ordered list (by angle) of polar coordinates (r, θ) where obstacles may be found. We suppose that θ is an error free coordinate. The line fitting is done in two steps. First we find clusters of points where the distance between two consecutive points is similar. Then we take advantage of the error-free coordinate and apply the transformations $u = \cos(\theta)/\sin(\theta), v = 1/r \sin(\theta)$ as in [11]. We find fit lines by applying to each cluster a divide-and-conquer technique combined with a minimal squares method.

Although, the minimal squares technique has the advantage of removing noisy measurements, it is not efficient in the number of lines it generates. For this reason a divide-and-conquer algorithm is necessary. We convert the generated vertices in the (u, v) space to a Cartesian space. Then, we apply a classical divide-and-conquer recursive technique to the vertices of each cluster to find the lines that fit the set of vertices, and by this, eliminating the unnecessary ones. A cluster with a stand alone point or with very few points should be considered as a small object [11], a sensor error or the result of a small free space between two occlusions. In any case, those should not be taken into account when the divide-and-conquer algorithm is applied.

The lines generated are considered as full edges, while the line that may be formed between two consecutive clusters is considered as a free edge.

3.3 Model-matching

The partial Hausdorff distance is used to find the best alignment between the previously explored region and the new one. The Hausdorff distance is computed on the original laser data of the polylines previously computed.

Given two sets of points P and Q , the Hausdorff distance is defined as (see [10]):

$$H(P, Q) = \max(h(P, Q), h(Q, P))$$

where

$$h(P, Q) = \max_{p \in P} \min_{q \in Q} \|p - q\| \quad (2)$$

and $\|\cdot\|$ is a norm for measuring the distance between two points p and q . The function $h(P, Q)$ (distance from set P to Q) is a measure of the degree in which each point in P is near to a point in Q . A small value of $h(P, Q)$ implies that every point in P is close to a point in Q . The Hausdorff distance is the maximum among $h(P, Q)$ and $h(Q, P)$. Thus the Hausdorff distance measures the degree to which each point of P is near a point in Q and vice versa.

By computing the Hausdorff distance in this way we are obtaining the most mismatched point between the two shapes compared, consequently, it is very sensitive to the presence of any outlying points. For this reason it is often appropriate to use a more general measure, which replaces the maximization operation with the calculation of the mean of the values. This measure (partial Hausdorff distance) is defined as:

$$h_k = M_{p \in P} \min_{q \in Q} \|p - q\| \quad (3)$$

Where $M_{p \in P} f(p)$ denotes the statistical mean value of $f(p)$ over the set P .

One interesting property of the Hausdorff distance and the ‘‘partial Hausdorff distance’’ is the inherent asymmetry in the computation. The fact that every point of P (or subset of P) is near some point of Q says nothing about whether every point of Q (or subset of Q) is near some point of P . In other words, $h_{k1}(P, Q)$ and $h_{k2}(Q, P)$ can attain very different values. In fact each one of the two values give different information.

The term $h_{k1}(P, Q)$ is the unidirectional partial distance from the previously explored region to the current perception, and $h_{k2}(Q, P)$ is the unidirectional partial distance from the current perception to the previously explored region. Where $P = M_t$ is the model and $Q = I_t$ is the model or region of the model given an t possible translation and rotation. The maximum of these two values defines the partial Hausdorff distance. The partial Hausdorff distance is function of a transformation composed by translation and rotation. The transformation that maximize the metric will determine the best alignment.

Map Building algorithm The general Map Building algorithm is as follows:

EXPLORATION(\mathcal{R}, \mathcal{W})

1. $P_n \leftarrow \text{TAKE_PERCEPTION}(\mathcal{R}, \mathcal{W})$
2. $\mathcal{M}.\text{add}(P_n)$
3. $U_f \leftarrow \text{ID_UNXP_AREAS}(\mathcal{M})$
4. **if** $U_f \neq \emptyset$ **then**
 - (a) $[x, y, \theta] \leftarrow \text{SELECT_NEW_POS}(U_f, \mathcal{R}, \mathcal{M})$
 - (b) **if** $\text{NOVALID}(x, y, \theta)$ **then**
 - i. Return \mathcal{M} , NOK
 - (c) $\mathcal{R}.\text{move}(x, y, \theta)$
 - (d) **goto** 1
5. **else** Return \mathcal{M} , OK

SELECT_NEW_POS($U_f, \mathcal{R}, \mathcal{M}$)

1. **for each** $u_f \in U_f$ **do**
 - (a) $l_n \leftarrow \text{RDNSAMPLE}(n, u_f, \mathcal{R}, \mathcal{M})$
 - (b) $L_q = L_q \cup l_n$
2. $q_{nbv} \leftarrow \text{MAXIMIZE}(\mathcal{T}, L_q)$
3. Return q_{nbv}

Where \mathcal{R} is the robot, \mathcal{W} the workspace, \mathcal{M} the map, l_n is the list of random positions per free edges and \mathcal{T} is the equation 1

4 Multi-Robot Map Building

We have developed, implemented and simulated a planner for multi-robot map building. Our approach consists in a centralized planner. The position and current visibility from every robot is assumed to be known by the planner. The main idea of this planner is that the robots must share the work of building a map.

Let us denote $V(q_k)$ as the visibility region of the robot k from the location q , $C_{r-}(V(q_k))$ as the visibility polygon reduced by the robot radius. $C_{r-}(V(q_k))$ is a safe region for navigation. $E_{fr}(q_k)$ denotes a free edge belonging to the visibility region $V(q_k)$, $V \cup (q_K)$ is the union of the visibility regions of all the K robots, $C_{r-}(V \cup (q_K))$ is the union of all $C_{r-}(V(q_k))$ for the K robots and $S_\tau(V \cup (q_K))$ is the state of the map at the time t . A path $\tau : [q_{init_k}, q_{goal_k}(E_{fr}(t_n))]$ will be admissible if $\tau : [q_{init_k}, q_{goal_k}(E_{fr}(t_n))] \in C_{r-}(V \cup (q_K))$. Where q_{goal_k} denotes the location of the robot k from where the free edge $E_{fr}(t_n)$ is seen and t_n denotes the position from where the robot n has computed this free edge.

In our multi-robot map building strategy every robot k has as a priority to visit a free edge $E_{fr}(q_k) \in C_{r-}(V(q_k))$ all the paths $\tau \in C_{r-}(V(q_k))$ are admissible. If a given robot k does not have any free edge to visit left in $V(q_k)$, this robot is allowed to explore a free edge $E_{fr}(t_n) \in V \cup (q_K)$ if there

is an admissible path $\tau : [q_{init_k}, q_{goal_k}(E_{fr}(t_n))]$. Associate each robot with its visibility region reduces the complexity of the problem. To build the map is necessary to send the robots to explore unknown space. When a robot k does not have any free edge to visit left belonging to its visibility region a combinatorial algorithm determines which free edge $E_{fr}(t_n) \in V \cup (q_K)$ must visit the robot by evaluating equation 1. In order to estimate the size of the unexplored space we are using the size of the free edges. Therefore, to explore the unknown space is equivalent to send a robot to visit a free edge.

We are not dealing with simultaneous robots motion, only a robot is moving at a given time the other robots remain motionless. A state $S_\tau(V \cup (q_K))$ changes to $S_{\tau+1}(V \cup (q_K))$ when the robot k has reached the location q_{goal_k} .

5 Robot Architecture and Experiments

We are using a Pioneer mobile robot (See figure 2) with an on-board PC 400 MHz processor. It is equipped with a Sony EVI-30 CCD moving camera for landmark identification. The robot is also equipped with a Sick laser range sensor. This sensor uses a time-of-flight technique to measure distances.

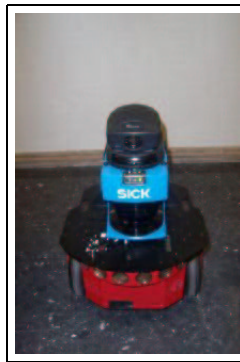


Figure 2: Indoor mobile robot and sensors

The software consists of several modules executing specialized functions and communicating using TCP/IP socket communications under a client/server protocol. The main modules in our robot architecture are: Frame server, sick laser server, line fitting module, model matching module, landmark identification server, motion planner, motion controller, and system coordinator.

We are currently developing and integrating the robot architecture necessary to perform our approach in a real robot. Up to now we have totally developed the frame server, motion planner, line fitting, sick

laser server, model matching, motion controller and system coordinator modules. We are working on the landmark identification module.

A computer simulation of this planner has been done. The software is written in C++ and uses geometric functions available in the LEDA 4.2 library. The simulation shows that this approach produces good results for the model building task.

In our simulation landmarks are represented with dark discs, the robot with a light square and the road-map with lines. The robot is placed anywhere inside the map, and begins exploring. As the robot moves across the map it takes every visibility area from the positions selected by the utility function to construct the model incrementally. The road map is constructed at the same time. The final map is constituted by polygons (which represent walls or obstacles), landmarks, and a road-map, constituted by a graph. When the robot ends exploring an area it is capable to go back since it remembers past unexplored areas. This *backtracking* is based on navigation across the graph.

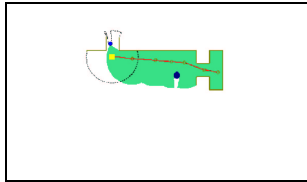


Figure 3:

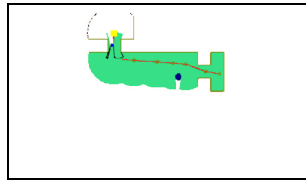


Figure 4:

Figures 3 and 4 show how the metric works: in figure 3 the robot has to take a decision between going to a large free edge, which means seeing as much of the as-yet-unseen environment as possible or going to a landmark to re-localize itself. In our simulation, the robot chose to improve its localization by going to the landmark (figure 4) and then go back and explore the unknown environment. In these figures the current visibility region is showed by a dotted line semi-circle.

Figure 5 shows an experiment on the real robot. The robot has chosen to go to the free edge in the front left because it represents the next best view according to the equation 1. We remark that the robot has not chosen to turn around to see the free edge behind it, because this configuration has a low value according to the equation 1. Actually, this configuration is bad because it is not possible to perform model matching and therefore deal with robot localization. Figure 6 shows the environment map at time t and the robot location at time t and $t + 1$. Figure 7 shows the environment map and robot location at

time $t + 2$. In both cases, the visibility robot region is shown in yellow, the free edges in red and the full edges in blue. Figure 8 shows the laser data at time $t + 2$ and figure 9 shows the robot going to the next best view.

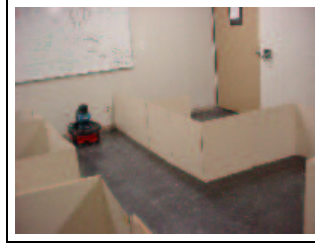


Figure 5: Experiment on a real robot

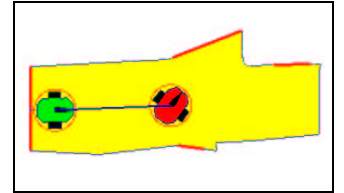


Figure 6: Environment map at time t

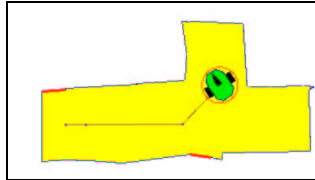


Figure 7: Environment map at time $t + 2$

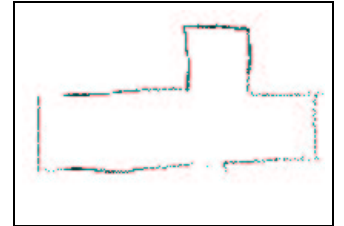


Figure 8: Laser data



Figure 9: Robot going to the next best view

Figures 16 and 17 show multi-robot map building. The colors in the map are used to associate a part of the map to the robot that has explored it. It can be seen that the map is uniformly distributed among the robots in the environment. The landmarks are shown in the figure as a blue disk (a column) or blue segments on the walls which represent posters. In figure 16 the environment is explored using non-limited range sensor and a 360 deg. visibility capability. It can be seen that with such conditions, the number of created milestones for sensing operations is smaller and the robot trajectory much simpler and shorter than in figure 17 where a limited range sensor and a visibility of 180 deg. was chosen.

Currently our global architecture is not complete yet. For the experiments performed on the robot, we are

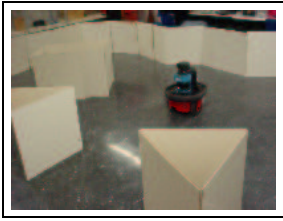


Figure 10:

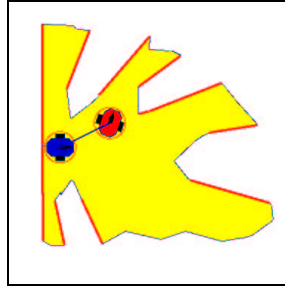


Figure 11:

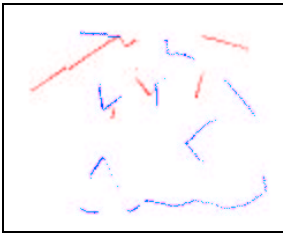


Figure 12:

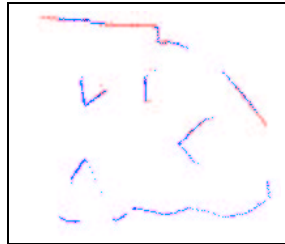


Figure 13:

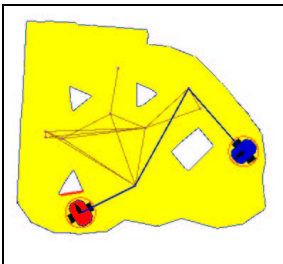


Figure 14:

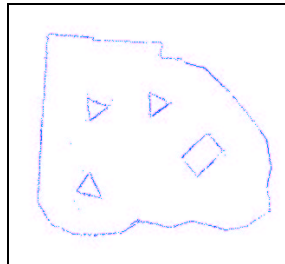


Figure 15:

using only data obtained from the laser. The camera is not used.

Figure 10 shows a picture of a more complex map building experiment. Figure 11 shows the map built at time t , the robots visibility region is shown in yellow, the free edges in red, and the full edges in blue. In this figure it is also shown the robots location a time t (blue disk) and the next position where the robot has to move at time $t + 1$ (red disk). This location has been computed using equation 1, but taking into account only the parameters related to the laser data.

Figure 12 shows the laser data at time t and $t + 1$ without registration. Laser data obtained at time t are in blue and those obtained at time $t + 1$ in red. Figure 13 shows the map matching obtained by using equation 2. The transformation (rotation and translation) related to the matching is used to improve

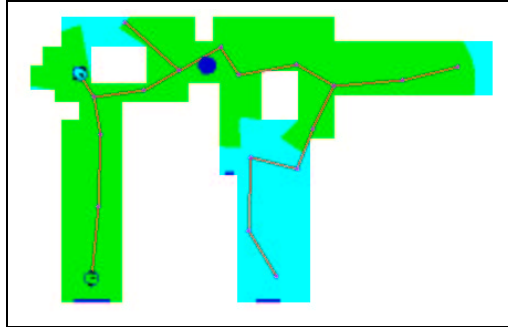


Figure 16: multi-robot map building: case of omnidirectional and infinite range

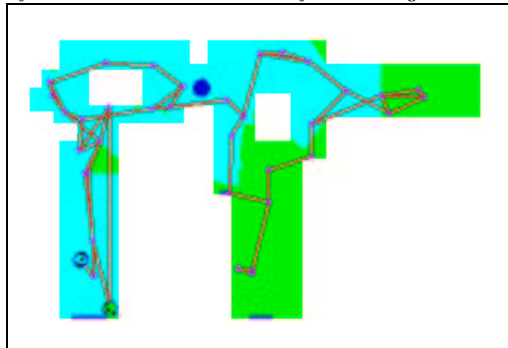


Figure 17: multi-robot map building: case of 180 deg. field of view and limited range

robot localization. Figure 14 shows the polygonal model of the environment and the last two robot locations. The road-map used by the robot is shown in the figure by using brown dotted lines. When the robot gets back to a previous location using the graph a localization procedure is executed at each graph node by using the model matching result. Figure 15 shows the laser final map.

6 Conclusions and Future research

A planner that selects the next position of the robot based on maximizing the utility function is proposed. The evaluation of this function uses the concept of robot information space which combines geometric information with an intensive usage of the results obtained from perceptual algorithms. The crux of our method is a randomized motion planner algorithm that, given a partial map of the environment, selects where to move the robot next. We balance the desire to see as much of the as-yet-unseen environment as possible, while at the same time having enough overlap and landmark information with the scanned part of the building to guarantee good registration and robot localization.

The final result of the exploration is a multi-representational map constituted by polygons, landmarks and a road-map.

Our approach is better than previous methods because the robot plans the motions in such a way that its uncertainty localization is minimized by using data registration and landmarks information. At the same time, the motion strategy take into account that the robot must move to discover unexplored environment regions by minimizing energy consumption. The robot motion strategy proposed generates a fast and reliable map building.

As future research, we will complete our global architecture and perform multi-robot map building experimentation. The representation of the environment proposed here is useful to several robotics task specially visibility-based ones such as target tracking and target finding. We believe that the model proposed can be used to select “good” locations where to perform 3D sensing operations. A 3D model of the environment is a inter-medium goal to other tasks (object manipulation, assembling, etc). This 3D modeling is one our possible future researches.

Acknowledgments

The authors thank Héctor González Baños for his contribution to the ideas presented in this paper. This work was funded by CONACyT project J34670-A and by the ITESM Campus Ciudad de México.

References

- [1] N. Ayache and O.D. Faugeras. Building, registration, and fusing noisy visual maps. *IEEE Journal of Robotics Research*, 7(6):45–65, 1988.
- [2] W.K. Lee B. Kuipers, R. Froom and D. Pierce. The semantic hierarchy in robot learning. In *IEEE Int. Conf. on Robotics and Automation*, 1993.
- [3] R. Chatila and J.P. Laumond. Position referencing and consistent world modeling for mobile robots. In *IEEE Int. Conf. on Robotics and Automation*, 1985.
- [4] H. Choset and J. Burdick. *Sensor based Motion Planning: The Hierarchical Generalized Voronoi Diagram*. In Algorithms for Robotic Motion and Manipulation, J.P. Laumond and M. Overmars (eds.), A K Peters, Wellesley (MA), 46-61, 1997.
- [5] J.L. Crowley. World modeling and position estimation for a mobile robot using ultrasonic ranging. In *IEEE Int. Conf. on Robotics and Automation*, 1989.
- [6] H. Bulata and M. Devy. Incremental Construction of Landmark-based and Topological Model of Indoor Environments by a Mobile Robot. In *IEEE Int. Conf. on Robotics and Automation*, 1996.
- [7] A. Elfes. Sonar-based real world mapping and navigation. *IEEE Journal on Robotics and Automation*, 3(3):249–264, 1987.
- [8] L.J. Guibas, J.C. Latombe, S.M. LaValle, D. Lin and R. Motwani. Visibility-based pursuit-evasion in a

polygonal environment. In *5th Workshop on Algorithms and Data Structures*, 1997.

- [9] J. Gonzalez, A. Reina and A. Ollero. Map building for a mobile robot equipped with a 2d laser rangefinder. In *IEEE Int. Conf. on Robotics and Automation*, 1994.
- [10] D.P. Huttenlocher et. al. Comparing images using the hausdorff distance. *IEEE Transactions on pattern analysis and machine intelligence*, 15(9):850–863, September 1993.
- [11] H.H. Gonzalez, E. Mao, J.C. Latombe and T.M. Murali. Planning robot motion strategies for efficient model construction. In *Robotics Research - The 9th Int. Symp*, 1999.
- [12] S. Lavelle, H. H. González-Banos, C. Becker, and J. C. Latombe. Motion strategies for maintaining visibility of a moving target. In *Motion Strategies for Maintaining Visibility of a Moving Target*, volume 1, pages 731–736, april 1997.
- [13] A. Lazanas and J.C. Latombe. Landmark-based robot navigation. *Algorithmica*, 13:472–501, 1995.
- [14] T.S. Levitt, D.M. Chelberg, D.T. Lawton and P.C. Nelson. Qualitative navigation. In *DARPA Image Understanding Workshop*, 1987.
- [15] R. Murrieta-Cid, M. Briot and N. Vandapel. Landmark identification and tracking in natural environment. In *proc International Conference on Intelligent Robots and Systems IEEE/RSJ-IROS’98*, Victoria, Canada, September 1998.
- [16] R. Murrieta-Cid, C. Parra, M. Devy, B. Tovar and C. Esteves. Building Multi-Level Models: From Landscapes to Landmarks. In *proc International Conference on Robotics and Automation 2002 (IEEE-ICRA’2002)*, Washington, USA, May 2002.
- [17] R. Murrieta-Cid, H.H. González and B. Tovar. A Reactive Motion Planner to Maintain Visibility of Unpredictable Targets. In *proc International Conference on Robotics and Automation 2002 (IEEE-ICRA’2002)*, Washington, USA, May 2002.
- [18] J. O’Rourke. *Visibility*. Handbook of Discrete and Computational Geometry, 467-479, J.E. Goodman and J. O’Rourke, 1997.
- [19] B. Tovar, R. Murrieta-Cid, and C. Esteves. Robot motion planning for model building under perception constraints. In *proc 9th International Symposium on Intelligent Robotic System (SIRS’2001)*, ages 447-456, Toulouse, France, July 2001.
- [20] S. Thrun, W. Burgard and D. Fox. Probabilistic mapping of an environmet by a mobile robot. In *proc IEEE Int. Conf. on Robotics and Automation*, 1998.
- [21] S. Teller. Automated urban model acquisition: Project rationale and status. In *DARPA Image Understanding Workshop*, 1998.