Exploration Strategies for Mobile Robots *

Camillo J. Taylor David J. Kriegman

Center for Systems Science
Department of Electrical Engineering
Yale University
New Haven, CT 06520-1968

Abstract

The problem of programming a robot to carry out a systematic exploration of its environment using realistic sensors is considered in this paper. The robot is modelled as a single point moving in a two-dimensional configuration space populated with visually opaque and transparent obstacles. The robot is equipped with proximity sensors, a vision-based recognition system, and a method of odometry, all of which have some uncertainty associated with their measurements. By using visually distinctive configurations of features in the world as natural landmarks, a series of local maps is constructed. The geometrical relationships between mutually visible landmarks are used to build a relational map from this collection of local maps. This novel representation forms the basis for a systematic exploration algorithm. The approach has been implemented in simulation, and results are presented.

1 Introduction

Advanced mobile robots in applications such as planetary exploration, construction, toxic waste cleanup, office automation and even domestic servitude will be called on to search for recognizable objects such as a mineral deposit, a particular I-beam, a leaking barrel of dioxin, a stapler or a lost baby's toy. To perform these tasks, the robot must be capable of systematically searching its environment using realizable sensing strategies.

In this paper, we consider the problem of robot exploration in a previously unknown environment. This task involves a number of sub-problems that have often been considered in robotics including motion planning, sensing, and uncertainty management. However, in most approaches to motion planning [12], it is assumed that an accurate representation of the environment is known. Lumelsky's work is an exception in that the motion planning strategies will achieve a goal without any prior description of the obstacles [17]. However, this work assumes that odometry is perfect and that the goal location is precisely known. Unfortunately, in exploration tasks, the goal location is unknown by definition, and it is unrealistic to believe that the *absolute* location of a robot can be precisely determined because of cumulative odometry errors and the compounding of uncertainty from external sensor measurements [21].

The approach presented in this paper will be based on local representations, and all motions will be specified with respect to visually distinct and recognizable objects which we will term *natural landmarks.* A search strategy is developed which will find a recognizable object in the presence of both odometric and sensing uncertainty. In order to cope with these uncertainties, new algorithms for building representations of the environment are presented along with algorithms that use these representations to plan and execute paths from place to place in the environment. The basic approach is to divide the world up into a number of overlapping regions, each of which can be represented by a local map, and to record the relationships between these regions.

Relational maps of various types have been proposed before, though these are typically based on a qualitative representations of different places and actions that link them [6, 11, 14, 18]. Additionally, information from multiple views can be integrated into a representation which acknowledges and attempts to remedy the errors due to odometric and sensor uncertainty [1, 2, 5, 10, 21]. Motion planning algorithms that can account for odometric and sensing uncertainty during path execution have been presented [4, 13]. Finally, Motion planning with respect

*Support for this work has been provided in part by NSF Young Investigator Award, IRI-9257990 and a gift from the INMOS division of SGS-Thomson.

1050-4729/93 $3.00 © 1993 IEEE

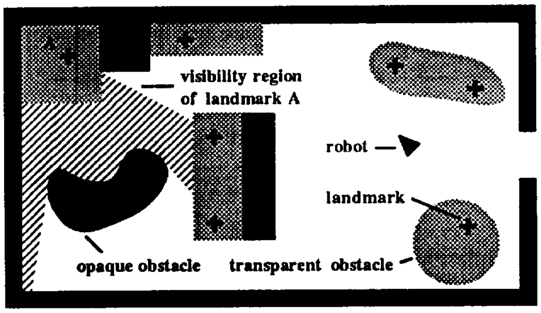

Figure 1: Major aspects of the robot's world model: Note that obstacles can take on arbitrary shapes and can be either transparent or opaque. The textured region indicates the visibility region of landmark A, that is, the region in freespace where this landmark is visible to the robot.

to a number of observable landmarks has been studied, but their locations must be known *a priori* [13, 14].

In the next section, the exploration problem is defined more precisely. In section 3, an algorithm for systematic exploration is presented. This algorithm employs a relational mapping system which it uses to plan paths in the presence of sensor uncertainty. This representation is discussed in section 4. The algorithm has been implemented in simulation and some experimental results demonstrating its behavior are shown in section 5. Finally, we are in the process of implementing this approach on our mobile robot, and our future directions are described in section 6.

2 Problem Formulation

In this paper, we will consider a circularly symmetric robot traveling in a planar world; this allows us to model the robot as a moving point in a two-dimensional configuration space (C-space). Figure 1 shows the major aspects of this world model. The triangle in the figure represents the position and orientation of a point robot, the shaded regions represent configuration space obstacles in the plane, and the crosses represent recognizable natural landmarks which will be defined below.

Following Lumelsky [17], the configuration space obstacles are modeled as simple closed contours which can take on arbitrary shapes. These obstacles come in two flavors, transparent and opaque. The opaque obstacles can occlude landmarks from the view of the robot; the transparent ones do not. In the real world, opaque obstacles might include walls and book shelves; transparent obstacles will block a robot's progress but the robot can either see through or over them (e.g. tables, windows, and wastepaper baskets). The obstacles and landmarks are assumed to be static, that is, their positions do not change over time.

The robot is equipped with a motion control system that accepts relative motion commands from the control program. There will be some error associated with the execution of each motion command, so we cannot obtain an accurate estimate for the absolute position of the robot by integrating these displacements. We also consider the robot to be equipped with a range sensor that can be used to perform boundary following. Such sensing can be achieved with an infra-red proximity sensor, a set of ultrasonic sensors or a tactile sensor (bumper).

The robot is also equipped with a recognition system which takes the image data obtained from a camera and returns the position and orientation of the robot with respect to any recognizable natural landmarks in the robot's field of view. A landmark is simply a group of features in the environment that can be reliably recognized by the vision system whenever it is in view. Various algorithms that can be employed to recognize objects and determine their position from image data are described by Grimson [7], Lowe [15], Ponce and Kriegman [9], Dhome et. al. [3] and others. One approach for recognizing landmarks would be to use one of the algorithms mentioned above to recognize specific objects from a library of models. Another approach would be to let the robot autonomously select groups of features that appear to be salient and recognizable.

Each landmark in this model is associated with a configuration space obstacle since every recognizable object must occupy a finite amount of space. Additionally, for each landmark we can define the region in freespace where that landmark is visible (see Figure 1). Following Lazanas and Latombe, we will refer to this area as the *visibility region* of the landmark [13].

3 Exploration Algorithm

The proposed exploration algorithm maintains a list of all landmarks that the robot has seen; a landmark in this list will be termed *visited* if the robot has circumnavigated the C-space obstacle containing the landmark. The goal of the exploration algorithm is to visit all of the landmarks in the environment. The algorithm repeatedly selects unvisited landmarks from this list, directs the robot toward the obstacle containing the landmark, and then circumnavigates this obstacle using sonar and tactile sensors. While the robot is executing this exploration algorithm it con-

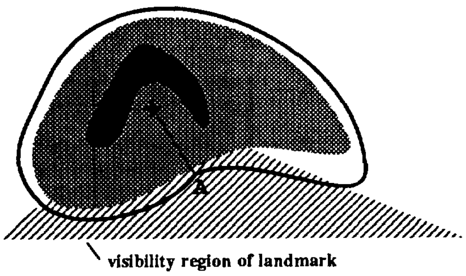

Figure 2: As the robot circumnavigates the C-Space obstacle it determines its closest approach to the landmark.

tinually collects sensor data which it uses to update its representation of the world and to determine which landmarks have been visited.

When searching for a particular object, the algorithm can be terminated as soon as it is found; since every landmark is eventually visited, the robot will either find the object or decide that it is not present. If odometry and sensing were perfect, then this algorithm would simply be a variation of the seed-spreader algorithm presented by Lumelsky, Mukhopadhyay and Sun for sensor based terrain acquisition [16].

In order to apply this algorithm we need to be able to use the sensor data to decide when a landmark has been visited. Consider the trajectory around the C-space obstacle shown in figure 2. Let **A** be the point in the visibility region of the landmark where the robot comes closest to that landmark. At that point the robot can decide whether the landmark is inside the current obstacle by considering the position of the landmark relative to the boundary of the C-space obstacle. If the boundary of the obstacle comes between the robot and the landmark at that point, then the landmark must lie inside the C-space obstacle, otherwise, it must be outside.

This observation provides us with a simple and robust procedure for deciding when a landmark has been circumnavigated; as the robot circumnavigates an obstacle it continually updates its records for its closest approach to each of the landmarks that it observes. When it has completed its circuit of the obstacle it can then decide which landmarks lie within the C-Space obstacle from these records. This procedure works quite well even in the presence of significant positioning error.

4 Sensors, Maps and Motion

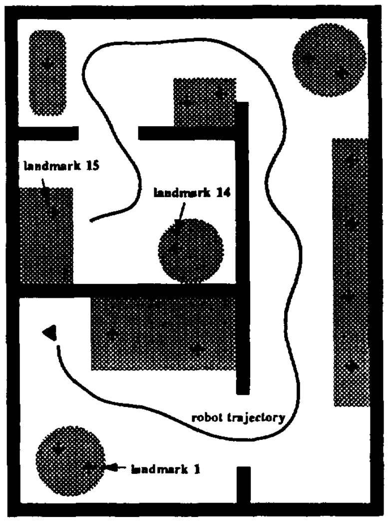

Figure 3: This figure shows the trajectory that a robot takes moving through a typical indoor environment

The exploration algorithm described in the previous section assumed that the robot was able to plan and execute paths through freespace. In order to do this, the robot must maintain a representation of the structure of its environment for use by a path planning algorithm.

In deciding how the robot is to represent its environment, we need to account for the characteristics of the sensor data that will be used to build and update this map. In particular, we need to consider the types of errors that can be expected from the sensors. We have assumed that the robot is equipped with a recognition system that can return the position and orientation of the robot with respect to every visible landmark. We can expect that the error in these position estimates will grow as the distance from the landmark increases. For example, several investigators have shown that the errors in positioning measurements from stereo systems grow quadratically with the distance to the feature [8, 10, 19]. Similar error analyses can be carried out for other feature-based pose recovery algorithms.

To motivate the need for a relational representation, consider the environment shown in figure 3. As the robot in this figure moves through the world it tries to estimate its position, and the position of all the landmarks it encounters with respect to a global frame of reference attached to landmark 1. Once the robot moves outside the visibility region of landmark 1, it can no longer directly measure its position with respect to the global frame of reference. Instead, it must calculate its position with respect to one or more of the visible landmarks and use its estimates for the positions of these landmarks to determine its location in the global map. This means that the error in the robot's position estimate will be a function of the errors in the estimates for the positions of the landmarks used in this calculation.

When a new landmark is subsequently encountered, the current estimate for the robot's location is used to calculate the position of this new landmark in the global map; again, this new estimate will inherit any errors present in the robot's location estimate. As the robot moves further away from its start position, the errors in the global map will simply accumulate, and the representation will become increasingly inaccurate.

This can cause serious problems for path planning algorithms that assume accurate global representations of the environment. In this case, a global map is simply the wrong coordinate system in which to record the measurements since it does not allow the robot to represent the structure of the errors in these measurements. The robot in figure 3 may not be able to accurately estimate the position of landmark 15 with respect to landmark 1, but it will be able to estimate the relationship between landmarks 14 and 15, and this information may be more relevant to the robot's navigation task than the position of landmark 15 in an arbitrarily chosen global frame of reference.

4.1 Local Maps

This observation leads us to propose an alternative approach to map making; instead of estimating the positions of landmarks and obstacles in a single global frame of reference, we associate a local map with each landmark in the environment and then maintain estimates for the relationships between these maps.

Figure 4 shows the local map associated with a particular landmark. Following Elfes, this map takes the form of an inference or uncertainty grid [5]. The polar grid was chosen because it reflects the structure of the visibility region of the landmark. The map is divided into a number of cells as shown in the figure, and the robot uses the information from its sensors to estimate the properties of each cell. More specifically, there are *two* attributes associated with each cell: a real number which represents the probability that the cell is occupied (or conversely that it is in free space) and a second number that represents the probability that the landmark is visible from that cell. These two attributes allow the robot to record the freespace in

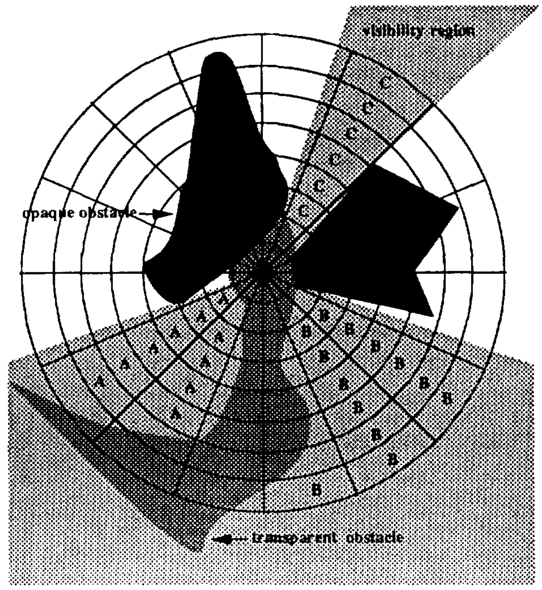

Figure 4: Local map of a landmark: The cells in this polar grid hold two values indicating whether they are in the visibility region of the landmark and whether they are occupied. This figure also shows how the visibility region of a particular landmark can be divided into three connected regions: A, B and C.

the local map and the visibility region associated with the landmark.

The information in the local maps can be used to divide the area around the landmark into distinct places as shown in figure 4. In this paper, a *place* is defined as a set of *connected* cells that are both in freespace and in the visibility region of a particular landmark. The most important property of a place is that the robot should be able to construct a path from any point in the place to any other point in the place without losing sight of the landmark (i.e. without leaving the visibility region).

4.2 Relational Maps

To determine the relationship between places in various local maps, we first need to understand the relationships between the landmarks. Whenever the robot has two or more landmarks in view, it can update its estimates for the relationships between them. In the planar world this relationship is uniquely specified by a translation vector, (t_x, t_y), and an angle, θ. In the current implementation, bounding intervals are used to express the uncertainty in these estimates [8] though other distributions might be appropriate.

From an estimate for the spatial relationship between two landmarks, it can be determined whether any of the places in the local map of one landmark coincide with any places in the other map. This can be done by calculating where each cell in the local map of landmark 1 would appear in the local map of landmark 2. The robot can then decide whether or not the cell is likely to be contained in any of the places associated with landmark 2. The robot takes into account the uncertainty in the estimate for the relationship between the landmarks when it performs this computation.

If the robot determines that a place in the local map of landmark 1 overlaps a place in the local map of landmark 2, it records the cells in the local map of landmark 1 that lie in the intersection of the two places and creates a connection between the two places. In this manner, the robot builds a directed graph of places that represents the structure of the environment.

4.3 Path Planning

Path planning occurs at two levels: at the global level, Dijkstra's algorithm is used to compute a path between the place that the robot is currently in and the place it wants to get to. This path takes the form of a series of connected places. Since each place is defined with reference to a particular landmark, the global plan can also be viewed as a sensor plan that specifies which landmark the robot should be observing at each stage.

The local planner is responsible for getting the robot between consecutive places in the global plan. It does this by planning a path through the occupancy grid of the current landmark to the intersection region of the two places.

Note that the robot can encounter previously undiscovered obstacles during its journey which may force it to replan at a local or global level.

5 Experimental Results

In order to test the algorithm described in this paper, a series of simulation experiments was carried out. These simulations were designed to capture the major features of the world model. The simulator provides the exploration algorithm with noisy sensor data from the vision and sonar sensors, and the exploration algorithm returns velocity vectors which are used to control the position and orientation of the robot.

The simulated recognition system returns estimates for the position of the robot with respect to all the landmarks that are visible from the robot's current position. These estimates are supplied in polar coordinates (r, θ) where r represents the distance from the landmark to the robot, and θ represents the orientation of the robot in the landmark's frame of reference. If r' is the true distance between the robot and the landmark, then the error in the estimate for r increases quadratically with r' while the error in the estimate θ increases linearly. For example, when the robot is 5 meters away from the landmark, the error in its estimate for r would be in the range ± 25 cm while the error in its estimate for θ would be in the range ± 10 degrees. The simulator also adds noise to the control vectors supplied by the exploration algorithm in order to reflect the kinds of problems one can expect with an actual mobile platform.

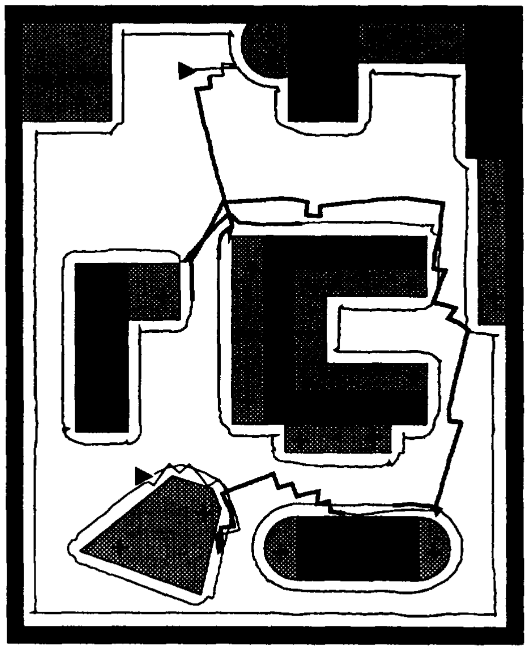

Figure 5: Path followed during a simulated exploration: The darker lines represent the portions of the trajectory where the robot was moving between two obstacles.

Figure 5 shows the trajectory of the robot in one of these simulation experiments. In this particular experiment the robot was told to explore every obstacle that contained a visible landmark represented by small crosses in this figure. Notice that the robot is able to successfully plan and execute paths that take it from one obstacle in the environment to another. For each landmark, an occupancy grid has been created, and the visibility regions have been determined. The place graph has been constructed and has been used in path planning. Note that the generated local path is always between the centers of the cells, and no path smoothing has been performed, so the path appears jagged.

6 Conclusion and Future Directions

We have presented in this paper an algorithm for systematically exploring a mobile robot's environment. Unfortunately, space limitations have precluded inclusion of important details such as explicit uncertainty models, the map updating scheme, and formal proofs. However, a number of issues still need further consideration. In this paper, the vision sensor is modelled as having full 360° coverage though most realistic sensors only cover a limited angular range. Assuming that constantly panning a camera over 360° is undesirable, sensor and motion planning will become more tightly coupled. It should be possible to plan paths that leave the visibility regions of all previously observed landmarks, yet are guaranteed to be able to return to a visibility region under some model of bounded uncertainty in odometry as in [13].

We are presently working towards implementing this approach on our mobile robot built on top of a TRC Labmate which hosts a network of onboard INMOS transputers for performing all vision and planning operations. A recently developed structure-from-motion algorithm will used to reliably determine the location of the straight line segments in a scene and the relative displacement of the robot [22]. Obstacle circumnavigation will be accomplished using a combination of tactile and sonar sensing.

References

[1] N. Ayache and O. Faugeras. Maintaining representations of the environment of a mobile robot. *IEEE Trans. on Robotics and Automation*, 5(6):804–819, December 1989.

[2] J. Crowley. Navigation for an intelligent mobile robot. *IEEE Journal of Robotics and Automation*, pages 31–41, Mar. 1985.

[3] M. Dhome, M. Richetin, J. Lapreste, and G. Rives. Determination of the attitude of 3-D objects from a single perspective view. *IEEE Trans. Pattern Anal. Mach. Intelligence*, 11:1265–1278, 1989.

[4] B. Donald and J. Jennings. Constructive recognizability for task-directed robot programming. In *IEEE Int. Conf. on Robotics and Automation*, 1992.

[5] A. Elfes. Sonar-based real-world mapping and navigation. *IEEE Journal of Robotics and Automation*, 3(3):249–265, June 1987.

[6] S. Engelson and D. McDermott. Image signatures for place recognition and map construction. In *SPIE Symp. on Intelligent Robotic Systems: Sensor Fusion IV*, 1991.

[7] W. E. L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* MIT Press, 1990.

[8] G. D. Hager. *Task-Directed Sensor Fusion and Planning : a computational approach.* Kluwer Academic Publishers, 1990.

[9] D. Kriegman and J. Ponce. On recognizing and positioning curved 3D objects from image contours. *IEEE Trans. Pattern Anal. Machine Intell.*, 1990. In press. An earlier version appeared in *Proc. Image Understanding Workshop*, Palo Alto, May 1989.

[10] D. J. Kriegman, E. Triendl, and T. O. Binford. Stereo vision and navigation in buildings for mobile robots. *IEEE Trans. on Robotics and Automation*, 5(6):792–803, December 1989.

[11] B. Kuipers and Y. Byun. A robust qualitative method for robot spatial reasoning. In *Proc. Am. Assoc. Art. Intell.*, pages 774–779, 1988.

[12] J.-C. Latombe. *Robot Motion Planning.* Kluwer Academic Publishers, Boston, 1991.

[13] A. Lazanas and J.-C. Latombe. Landmark-based robot navigation. In *Proc. Am. Assoc. Art. Intell.*, 1992.

[14] T. Levitt, D. Lawton, D. Chelberg, and P. Nelson. Qualitative navigation. In *Proc. Image Understanding Workshop*, pages 447–465, 1987.

[15] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, 1987.

[16] V. Lumelsky, S. Mukhopadhyay, and K. Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Trans. on Robotics and Automation*, 6(4):462–472, 1990.

[17] V. Lumelsky and A. Stepanov. Dynamic path planning for a mobile automaton with limited information on the environment. *IEEE Trans. on Automatic Control*, 31(11):1058–63, 1986.

[18] M. Mataric. Integration of representation into goal directed behavior. *IEEE Trans. on Robotics and Automation*, 8(3):304–312, June 1992.

[19] L. Matthies and S. Shafer. Error modeling in stereo navigation. *IEEE Journal of Robotics and Automation*, 3(3):239–248, 1987.

[20] J. Mundy and A. Zisserman. *Geometric Invariance in Computer Vision.* MIT Press, Cambridge, Mass., 1992.

[21] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[22] C. Taylor and D. Kriegman. Structure and motion from line segments in multiple images. In *IEEE Int. Conf. on Robotics and Automation*, pages 1615–1620, 1992.