Coastal Navigation – Robot Motion with Uncertainty

Nicholas Roy[1], Wolfram Burgard[2], Dieter Fox[1], Sebastian Thrun[1]

[1] School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

[2] Computer Science Department
University of Bonn
Bonn, Germany

Abstract

Ships often use the coasts of continents for navigation in the absence of better tools such as GPS, since being close to the land allows sailors to determine with high accuracy where they are. Similarly for mobile robots, in many environments global and accurate localization is not always feasible. Environments can lack features, and dynamic obstacles such as people can confuse and block sensors.

In this paper, we demonstrate a technique for achieving better localization by generating trajectories that model explicitly both the information content of the environment, and the density of the people in the environment. These trajectories reduce the average positional certainty along their length, reducing the likelihood the robot will become lost at any point.

Introduction

One essential component of any operational mobile robot system is the ability for the robot to localize itself; that is, to determine its position in space consistently and accurately from sensor data. Dead reckoning using only odometry data does not solve this problem; small errors in odometry build up quickly, eventually causing dramatic errors in the robot's belief of its position. Over distances longer than a few meters, the robot must use information from its environment to track where it is. There are many successful localization methods that can determine the robot's position (relative to a map) using sonar, laser and camera data (MacKenzie & Dudek 1994; Dudek & Zhang 1996; Sim 1998; Thrun, Fox, & Burgard 1998; Fox, Burgard, & Thrun 1998; Kaelbling, Cassandra, & Kurien 1996).

However, most localization methods fail under common environmental conditions. Proximity sensors such as laser or sonar range-finders have finite range, which means that in sufficiently wide-open spaces, they cannot see anything to use as a reference point. Such sensors can also be fooled by unmodelled or dynamic obstacles; people moving around the robot are a very good example of unmodelled, dynamic obstacles. Cameras can also fail in regions which lack sufficient visual structure, such as blank walls or ceiling. Since these environmental conditions are relatively common, a mobile robot navigating reliably in the real world must allow for the potential failure of its localization methods.

The inspiration for the method discussed in this paper stems from traditional navigation of ships. Ships often use the coasts of continents for navigation in the absence of better tools such as GPS, since being close to the land allows sailors to determine with high accuracy where they are. The success of this method results from coast lines containing enough information in their structure for accurate localization. By navigating sufficiently close to areas of the map that have high information content, the likelihood of getting lost can be minimized.

The coastal navigation technique consists of the following:

- We construct a model of the information content of the environment. This model allows us to account both for sensor limitations and unmodelled, dynamic obstacles.
- We plan a trajectory by combining the information content model of the environment and obstacle information in the map, to generate appropriate paths through the environment that respects localization uncertainty.

In the following sections, we first develop the coastal navigation model of the information content of the environment, starting with the sensor limitations and then accounting for dynamic obstacles. Secondly, we develop a method of combining the information content with the path planner to generate plans with minimum localization error. Finally we show experimental results.

The framework that we use for navigation is a probabilistic one; all information is represented as probabilities, such as the robot position and the environment map. Figure 1 shows an overhead (bird's-eye view) of an example environment. This map is a probabilistic occupancy grid, where each cell contains the probability that it is occupied. The black cells are those with a high probability of occupancy, such as cells inside walls. The white cells are cells in free space.

The sensors used to generate this map were 2 laser range finders which provide 360° field of view around the robot at 45cm height, with an angular resolution of 1°. The resolution of the map in figure 1 is 20cm/cell.

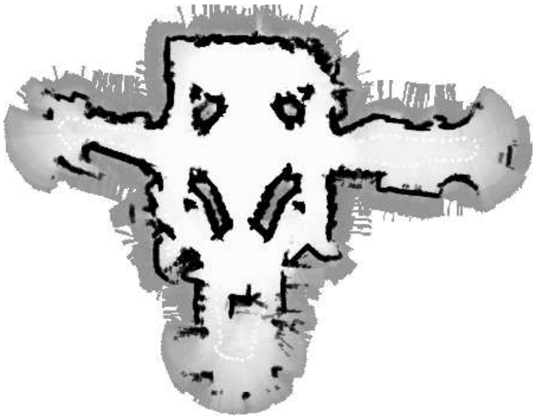

Figure 1: An example map of the National Museum of American History. The white areas correspond to open space, and the black areas are walls, or occupied space. The size of this map is 53m by 67m

This map is the National Museum of American History (NMAH), and was learned by the robot Minerva as part of a demonstration of robot technology (see (Burgard *et al.* 1998) for a description of a previous demonstration). Figure 2 shows Minerva, a RWI B-18 base, that was used for a two week period in the museum.

Figure 2: Minerva, the robot used for the experiments presented in this paper.

The NMAH was considered to be a good testbed for the idea of navigating with uncertainty, because it has two features relevant to developing coastal navigation: large areas with minimal environmental structure, and dynamic obstacles. The ideas were tested on the Minerva robot over the two week period, encountering approximately 50,000 people and travelling 44 km total in this environment. The area of the museum that Minerva operated in was 53m by 67m.

The success of this work will first be seen to be that the paths generated on a real robot, in a large, open and extremely dynamic environment (the museum) were generated as expected, and successfully followed. We have begun an in-depth evaluation of the data gathered over these trajectories, which is incomplete but promising. The main result presented in this paper is that of the successful operation of the coastal navigator in the adverse conditions, over a long-term period of two weeks.

Previous Work

Developing motion planning algorithms based on positional uncertainty is not a new idea. Considerable work in in the field of partially observable Markov decision processes (POMDPs) (Cassandra, Kaelbling, & Littman 1994; Koenig & Simmons 1996; Kaelbling, Cassandra, & Kurien 1996; Koenig & Simmons 1998) has allowed many mobile robots to model positional uncertainty explicitly. However, one drawback to the use of traditional POMDPs is that they can become computationally intractable with a large number of states.

Work has also been done on trajectory generation with respect to positional uncertainty; Takeda et al. (Takeda, Facchinetti, & Latombe 1994) do not use the localization process to generate the positional uncertainty across space, but generate probability distributions based on an explicit model of the sensor. Furthermore, the environment is assumed to be static, so the effect of dynamic obstacles on localization is not modelled.

Modelling Information Content

The motivation for coastal navigation is generating trajectories for the mobile robot that reduce the likelihood of localization error. An example of a trajectory where localization error is high, is a path through wide-open space, where all reference points are outside the range of the sensors – the likelihood of the robot becoming lost as it moves through the open space is high. We therefore need to identify how good regions of space are for localization, and alter the path planning accordingly. To do so, we build maps of the environment, where each cell in the map contains a notion of information content available at that point in the map, which corresponds to the ability of the robot to localize itself. The higher the information content, the better the localization ability.

The framework for determining the information content that we have used stems from the probabilistic model used by our localization method, which is a grid-based implementation of Markov localization (Burgard *et al.* 1996). This localization methods is able to represent the position of the robot using arbitrary discrete probability distributions. The probability of the robot having a particular pose (x, y, θ) has a probability $P(x, y, \theta)$.

Given a set of data (for example a set of range data from a laser sensor) our localization program returns a new probability distribution over the space of poses of the robot. The new probability function reflects whether or not the sensor data agrees or disagrees with the previous position estimate, and whether or not the sensor data increases or decreases the certainty of the robot's location.

We can determine with how much certainty the robot is localized by examining the entropy of the resulting probability function. The entropy of a probability distribution, P, is defined as:

$$Entropy(P) = -\sum_{P} p_i \log(p_i) \quad (1)$$

$$\Rightarrow Entropy(P) = -\sum_{X,Y,\Theta} p(x,y,\theta)\log(p(x,y,\theta)) \quad (2)$$

We sum over the space of all possible poses (x, y, θ), where the probability that the robot is at pose (x, y) is $p(x, y, \theta)$, a value provided by the localize module.

We can ignore the rotational component of the robot's pose, because we are assuming the use of Minerva, a robot with 360° field of view. Therefore, the sensor scan is rotationally invariant, and thus the θ term to the localization has been ignored to reduce the computational complexity of the approach.

This entropy quantity is defined for both the prior position estimate, and the posterior estimate provided by the localization module. The entropy can be considered as the "purity" of the probability distribution. If the distribution is highly focussed at a single point, (x, y) then the entropy will be low. If the distribution is spread over a wide space, then the entropy will be high. By examining the change in entropy, $\Delta(Entropy)$, we can determine whether or not the sensor data helped the robot in finding its position or not.

The change in entropy from the sensor reading can therefore be considered as the utility, or information content, of that sensor reading. By examining the sensor data that can be acquired at each point in space, we can determine the information available at each point in space and plan trajectories to maximize information available to the sensors, in the manner described below.

To construct the map of entropy, or information content of each position, we simulate sensor data for each (x, y) point in the map. This simulated sensor data is then used to localize the robot, which gives a probability distribution of the robot's pose. That is, given an initial position (x, y) of the robot, and a set of sensor data $\vec{s}$, the localization generates a probability distribution $P(x', y'|x, y, \vec{s})$, which we use to compute the *a posteriori* entropy as follows:

$$E(x,y) = -\sum_{X',Y'} p(x',y'|x,y,\vec{s})\log(p(x',y'|x,y,\vec{s})) \quad (3)$$

The entropy, $E(x, y)$, of the probability distribution that results from localization over sensor data, $\vec{s}$, simulated at (x, y) is computed. By assuming a Gaussian prior probability distribution centered at the assumed location of the robot, the entropy of the posterior probability distribution is equivalent to the change in entropy between the two.

Figure 3 shows an example map of the entropy, or information content of the same museum. The darker an area is, the less information it contains. Notice that the darkest area is the center of the large open space in the middle, and that the lightest areas, with the lowest entropy are close to the walls.

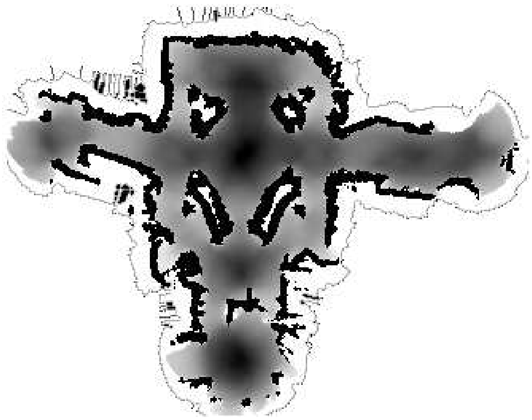

Figure 3: An example map of the entropy, or information content, of the National Museum of American History. The darker an area is, the less information content it contains. The blackest areas of the map are the walls.

It is important to note that in general, if the robot does not have rotationally-invariant perceptions, then this term cannot be ignored; indeed, coastal navigation is not very helpful if the ship is always pointed out to sea.

Modelling People

The entropy maps are useful for determining the information content of a particular point in the environment. However, the model used to compute the entropy assumes a static environment; in a dynamic environment, the data gathered by the sensors can be corrupted, for example by people blocking the proximity sensors. We therefore must also account for the likelihood that information can be corrupted.

We have two methodologies for accounting for corrupted data. The first is a brute-force simulation of people; the environment is assumed to have a certain number of people, uniformly distributed throughout the space, and this distribution of people is used to simulate corrupted data.

In the example of the laser range sensors, the probability that a given laser range measurement will be corrupted by a person is modelled as a geometric distribution along the length of the beam; the longer the beam, the more likely it will be corrupted.

Along each beam, the distance to the nearest person is generated randomly according to this distribution. If the person is closer along this direction than the nearest obstacle, the beam is corrupted. This simulated data is used to localize the robot in the map, as before. However, since the sensor data is not likely to exactly match any pose in the environment but only approximately, the certainty of the robot's belief is likely to be lower, and thus the entropy higher, than the uncorrupted sensor data generated for the same position.

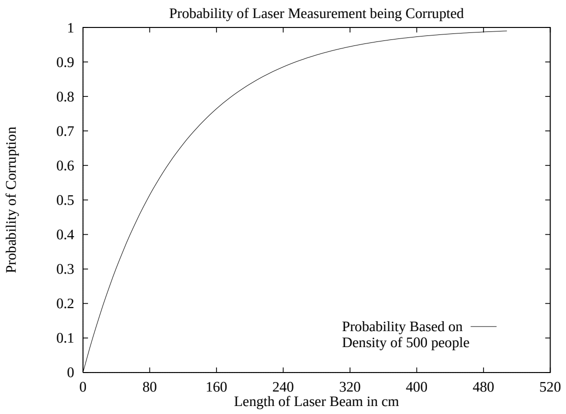

Figure 4: Probability of corruption of laser measurement, as a function of measured distance, given 2000 people in the museum.

Figure 4 shows an example distribution for the probability of a measurement's corruption, as a function of the measured distance. This distribution assumes that there are 2000 people, evenly distributed through a space the size of the NMAH. One effect of the model of people as discussed above is that, continuing the example of the laser range sensor, since longer beams are more likely to become corrupted than shorter beams, the entropy of a probability distribution generated by a set of data is likely to be higher if the laser beams are in general longer, rather than shorter. Thus, there is a trade-off between positions in the map that contain complete, but highly-corruptible information, compared with positions in the map that contain incomplete, but more reliable information.

One problem with the technique of simulating people to corrupt laser scans is that it relies on a randomly-generated simulation. This implies computing statistics over several trials to make any kind of substantial claims about the effect of people on the entropy, which is expensive in terms of computation time. Therefore, an alternative strategy has also been explored.

In this alternative strategy, again using the example of the laser range finder, each laser measurement is used to compute the entropy, and weighted by the probability that it is corrupted, which is a function of its length. Specifically, rather than localizing the robot over the entire laser data set, each single laser measurement is used to localize the robot, generating (for 360 laser measurements), 360 probability distributions. The entropy of each distribution is computed, weighted by the probability of the beam being *uncorrupted* – short beams get higher weight. The average of these weighted entropies is then computed, to give an overall measure for the (x, y, θ) location, as given in equation 4:

$$E_{av}(x,y) = \frac{1}{|\Theta|}\sum_{\Theta}(E(x,y,\theta)\cdot p_{corrupt}(x,y,\theta)) \tag{4}$$

In equation 4, $E_{av}(x, y)$ represents the entropy averaged over n different measurements (i.e., 360 laser measurements), whereas $E(x, y, \theta)$ is the entropy of the probability distribution given using only the single laser measurement at angle θ, and $p_{corrupt}(x, y, \theta)$ is the probability that the laser measurement taken along angle θ from position (x, y) is corrupted.

This particular method makes a strong independence assumption about the sensor data; it assumes that each laser measurement is completely independent of all others. This, in reality, is not the case. However, we can make reasonable conclusions about the information content of each position in the map nevertheless, and this method has the advantage of being computationally much faster than the previous method of modelling the effect of people on the sensors.

We in fact now have two different methods for computing the effect of dynamic obstacles, such as people, on the ability of the robot to localize itself. The first method is statistical in nature, and is more correct in representing the actual localization mechanism, however it is computationally slow. The second method makes a strong independence assumption about different sensor measurements taken at the same place, but yields equivalent results, and is an order of magnitude faster.

Path planning

Having computed the information content, or entropy, for each position in the map, the path planner must use the secondary map to generate trajectories with greater positional certainty. Traditional non-topological path planners choose a trajectory by optimizing some criterion such as minimizing distance, time, or power consumption, or maximizing distance to obstacles (for safety). The quantity minimized in the conventional planner is the following sum (Thrun 1999), along the path given by the list of cells (x_i, y_i) from start to goal:

$$Cost_{Total} = \sum_{x,y} c(x,y) \tag{5}$$

The cost $c(x, y)$ is the cost of crossing cell (x, y), which increases with the probability that the cell is occupied, from some minimum cost associated with travel. An example trajectory is shown in figure 5. The trajectory of the robot is the line through the large open space, where the start position is the left end of the line, and the goal is the right end. People are not depicted in this image, but typically, visitors to the museum would occupy the space on either side of the robot, effectively blinding it on its two sides, reducing substantially the main sources of localization information.

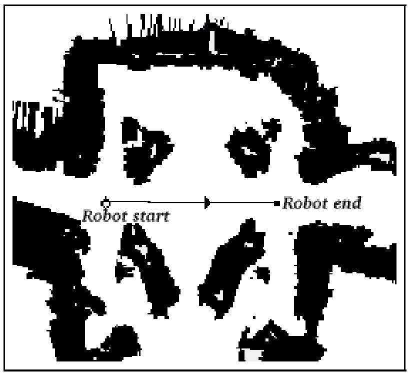

Figure 5: An example trajectory using the conventional planner, in the National Museum of American History.

The coastal planner, however, minimizes a sum of the conventional cost and the entropy:

$$Cost_{Total} = \sum_{x,y} \lambda_1 c(x,y) \cdot \lambda_2 E(x,y) \tag{6}$$

The exponents λ_1 and λ_2 are weights, and were chosen experimentally.

Figure 6 shows a coastal plan for the same start and goal as figure 5, where the robot does not travel directly through the open space, but instead hugs the wall, increasing travel distance, but preserving the ability to gather sensor data down its right side (travelling left to right again).

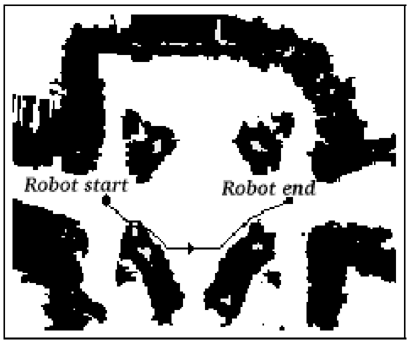

Figure 6: An example trajectory using the coastal planner, in the National Museum of American History. Note that the robot now hugs the wall.

Experimental Results

Over the course of two weeks, the robot gave tours of various exhibits scattered about the hall shown in figure 1, using the coastal planner to generate trajectories between exhibits. The sensor and localization data was recorded during this time, and some statistics were gathered to compare the performance of the coastal planner to the conventional planner.

The most useful statistic is the average entropy of the probability distribution of the robot's pose, as it travelled along the trajectories. In the best case, the robot followed trajectories that had a measurably lower average entropy, which indicates the success of the coastal navigation.

Coastal	Conventional
$3.3 \pm .1$	$4.4 \pm .25$

Table 1: Comparison of average entropy over the trajectories given by the conventional and coastal planners

Surprisingly, the largest impact of the coastal planner on positional uncertainty was seen in situations without the dynamic obstacles of surrounding people. People following the robot had a tendency to drive the robot off planned trajectories, making reasonable comparisons of the two planners very difficult. In the worst case, no difference in the average entropy was seen at all.

Conclusion

In this paper, we have presented a method of generating trajectories even through environments where positional uncertainty is likely to accrue. The method draws on ship-based navigation, where ships lacking reliable global position estimation stay close to known landmarks along shores. We first generate a map of the environment that contains the information content of each position in the environment. This representation also includes the likelihood of the sensor data to be corrupted. Using this map, the path planner generates trajectories that optimize over both distance and change in positional certainty. This path planner was used for navigating in a highly dynamic environment with large open spaces in the National Museum of American History successfully for 2 weeks.

One strength of the framework of coastal navigation is that it generalizes to any sensor; indeed, using a probabilistic localization module based on ceiling images generated by a camera, the path planner generated trajectories that did not stay close to obstacles. Instead, the path planner generated trajectories that took the robot under as much visual structure on the ceiling as possible, most notably the ceiling lights.

One avenue for future research lies with the path planner. The dynamic programming technique currently used for finding the minimum-cost trajectories demands a monotonic integration of the entropy. Therefore, there is no way to model actions that reduce uncertainty, only actions that increase uncertainty more or less.

References

Burgard, W.; Fox, D.; Hennig, D.; and Schmidt, T. 1996. Estimating the absolute position of a mobile robot using position probability grids. In *Proc. of the Thirteenth National Conference on Artificial Intelligence*, 896–901.

Burgard, W.; Cremers, A. B.; Fox, D.; Lakemeyer, G.; Hähnel, D.; Schulz, D.; Steiner, W.; and Thrun, S. 1998. The interactive museum tour-guide robot. In *Proc.of the Fifteenth National Conference on Artificial Intelligence.*

Cassandra, A. R.; Kaelbling, L. P.; and Littman, M. L. 1994. Acting optimally in partially observable stochastic domains. In *Proceedings of the Twelfth National Conference on Artificial Intelligence.*

Dudek, G., and Zhang, C. 1996. Vision-based robot localization without explicit object models. In *Proc. International Conference of Robotics and Automation.* Minneapolis, MN: IEEE Press.

Fox, D.; Burgard, W.; and Thrun, S. 1998. Active markov localization for mobile robots. *Robotics and Autonomous Systems.* to appear.

Kaelbling, L. P.; Cassandra, A. R.; and Kurien, J. A. 1996. Acting under uncertainty: Discrete bayesian models for mobile-robot navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems.*

Koenig, S., and Simmons, R. 1996. The effect of representation and knowledge on goal-directed exploration with reinforcement learning algorithms. *Machine Learning Journal* 22:227–250.

Koenig, S., and Simmons, R. 1998. *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems.* MIT Press. chapter A Robot Navigation Architecture Based on Partially Observable Markov Decision Process Models.

MacKenzie, P., and Dudek, G. 1994. Precise positioning using model-based maps. In *Proceedings of the International Conference on Robotics and Automation.* San Diego, CA: IEEE Press.

Sim, R. 1998. Mobile robot localization from learned landmarks. In *Proceedings of IEEE/RSJ Conference on Intelligent Robots and Systems (IROS).*

Takeda, H.; Facchinetti, C.; and Latombe, J.-C. 1994. Planning the motions of mobile robot in a sensory uncertainty field. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 16(10):1002–1017.

Thrun, S.; Fox, D.; and Burgard, W. 1998. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning 31 and Autonomous Robots 5 (joint issue)* 253–271.

Thrun, S. 1999. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence* 1:27–71. to appear.